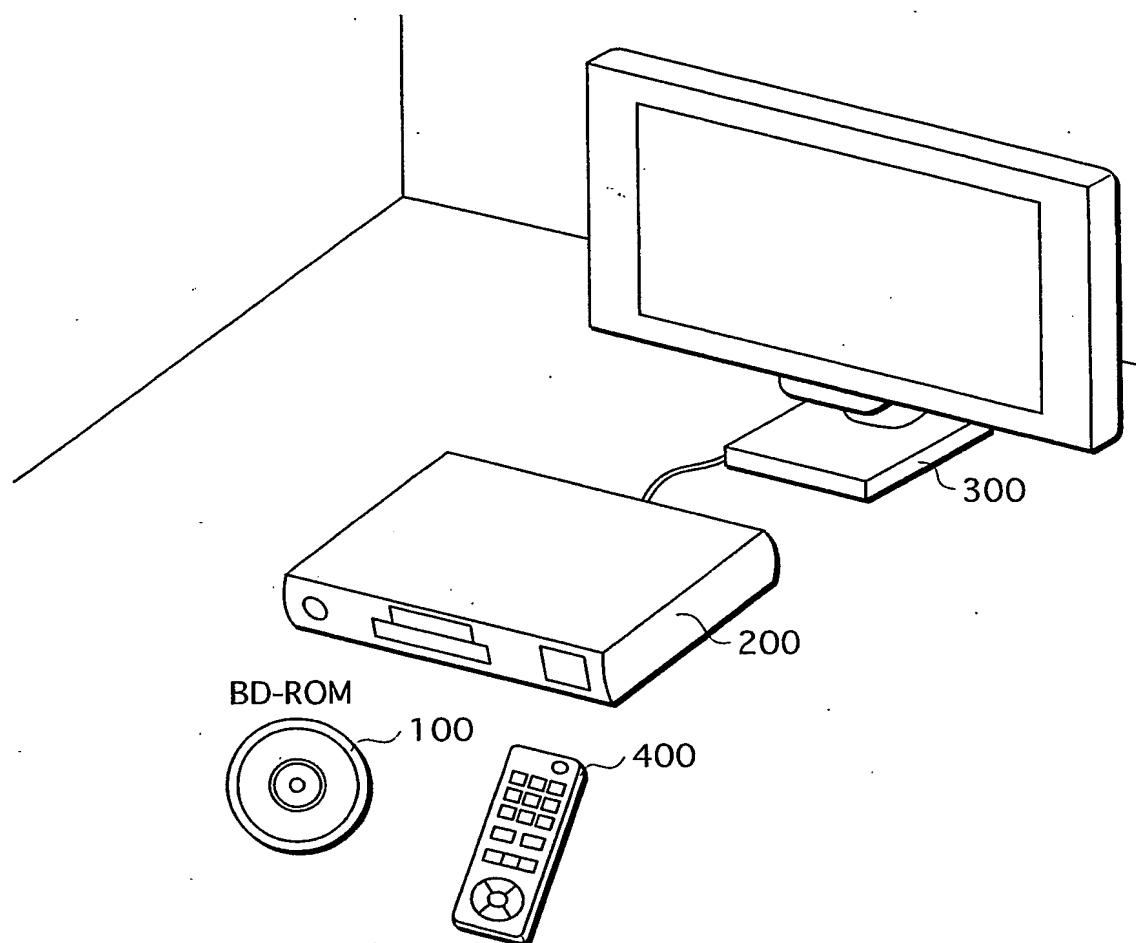
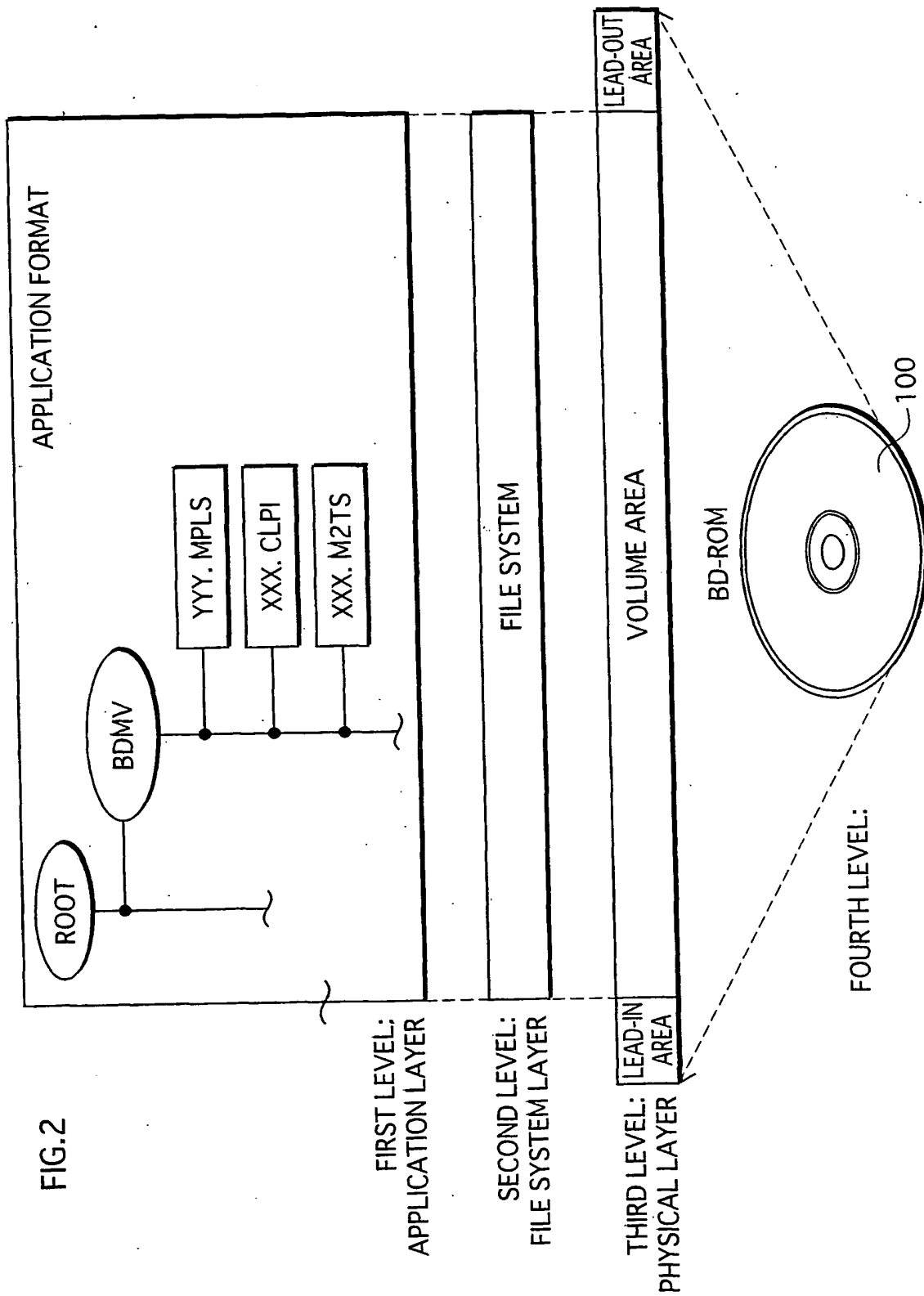
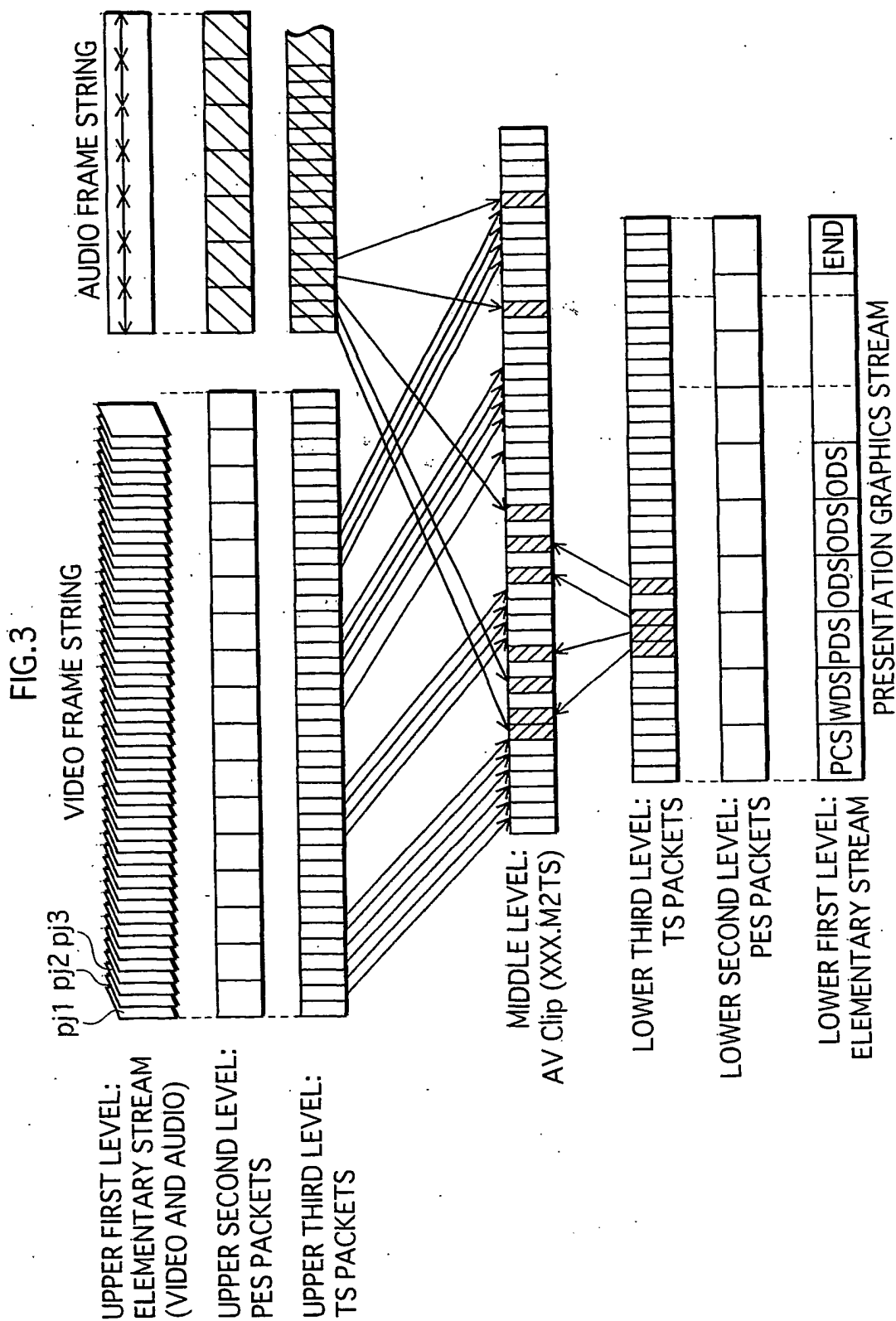


FIG.1







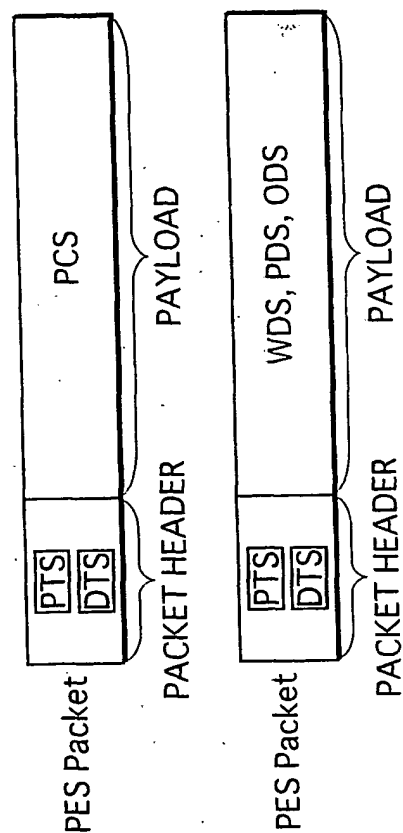
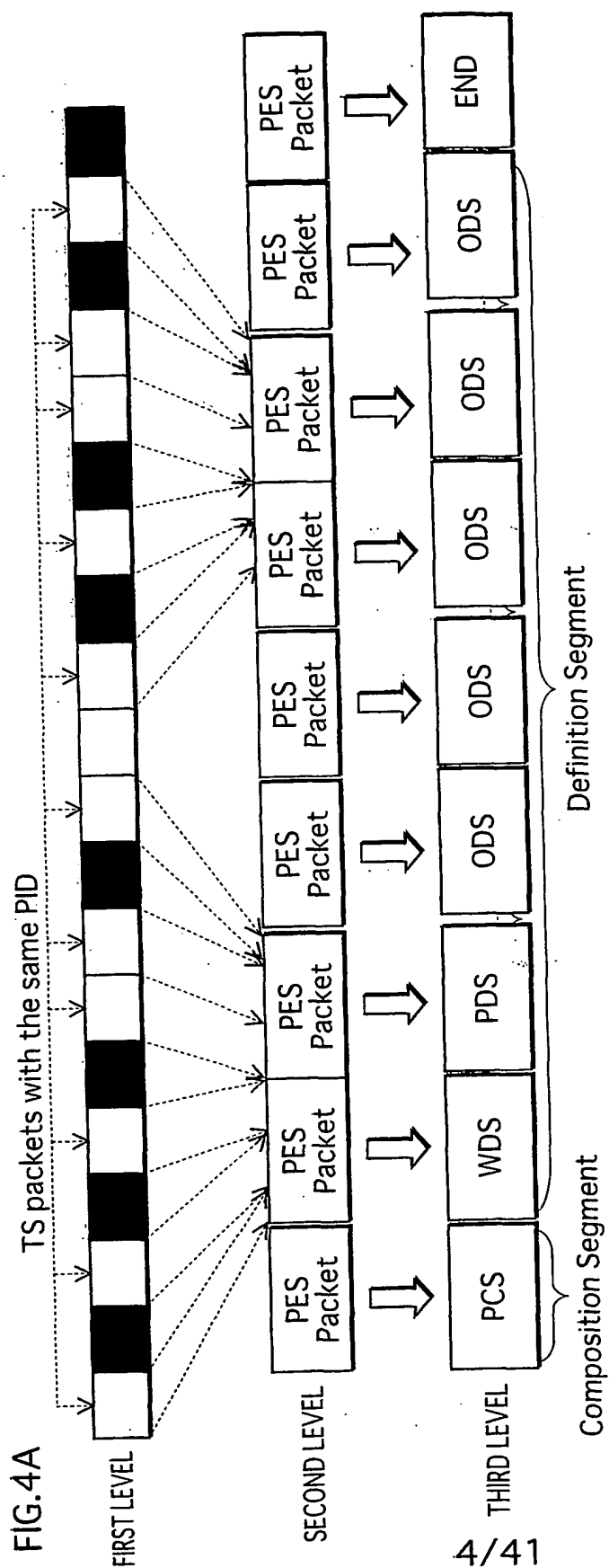


FIG. 4B

FIG.5

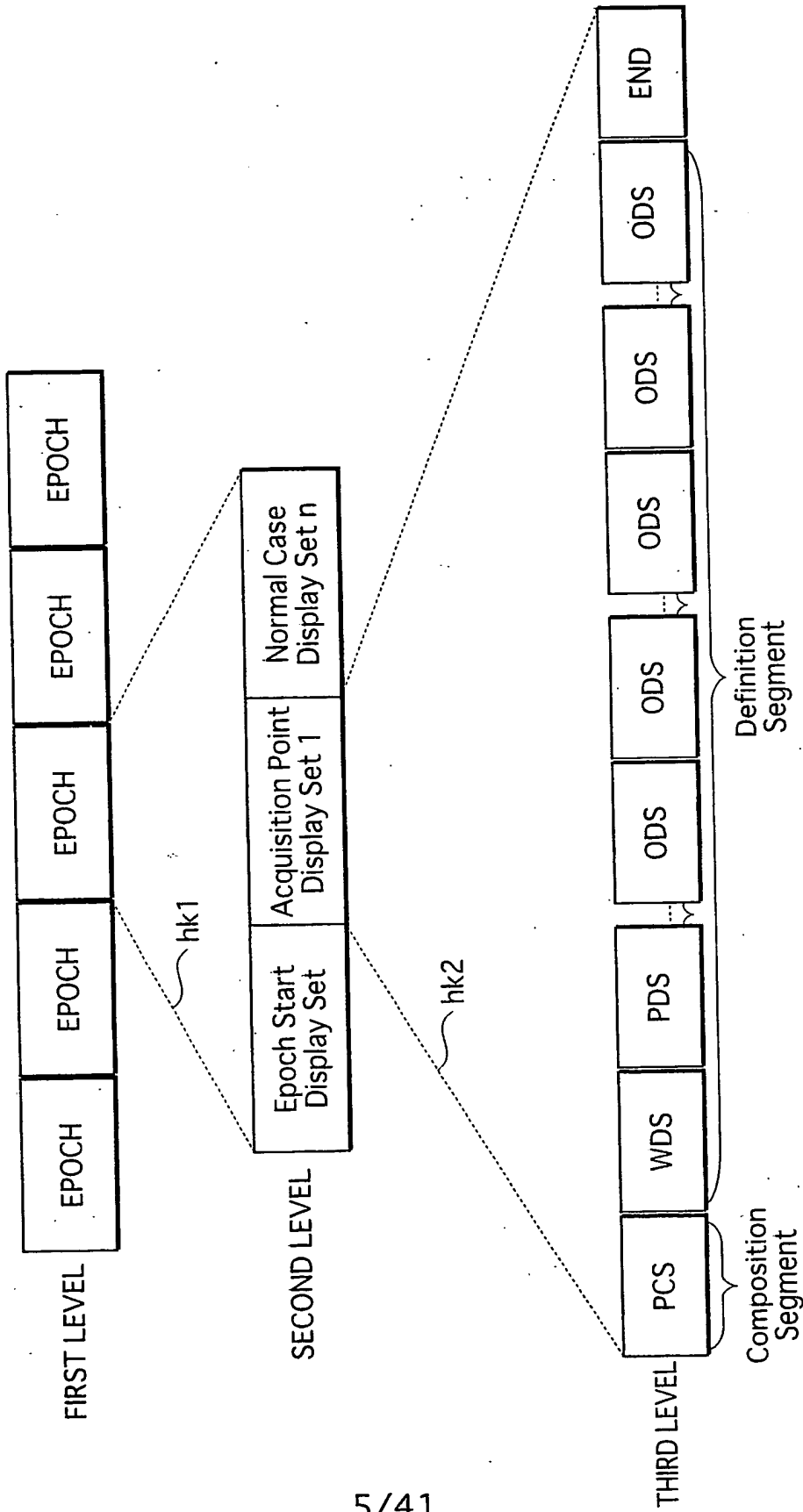


FIG.6

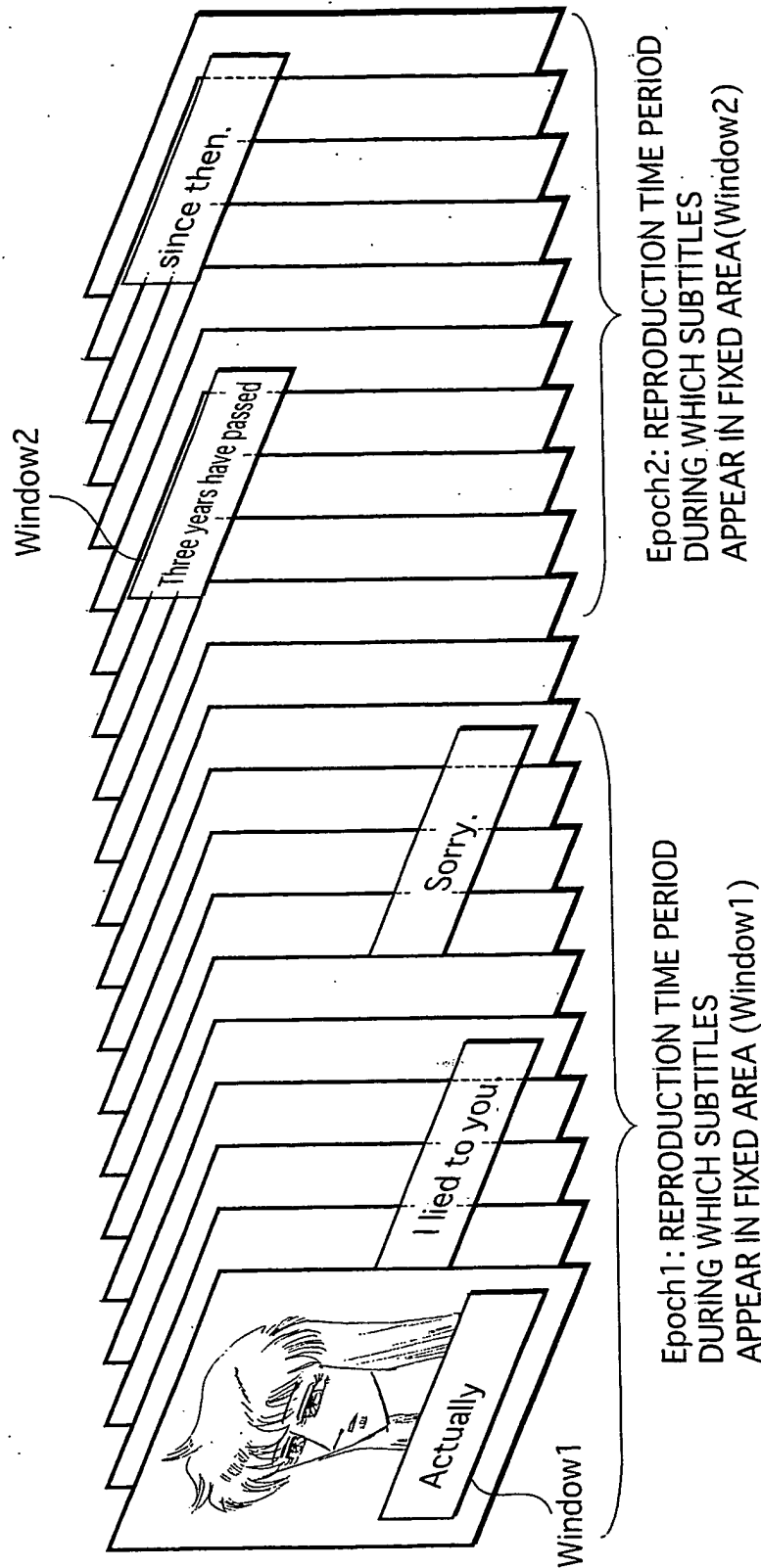


FIG.7A

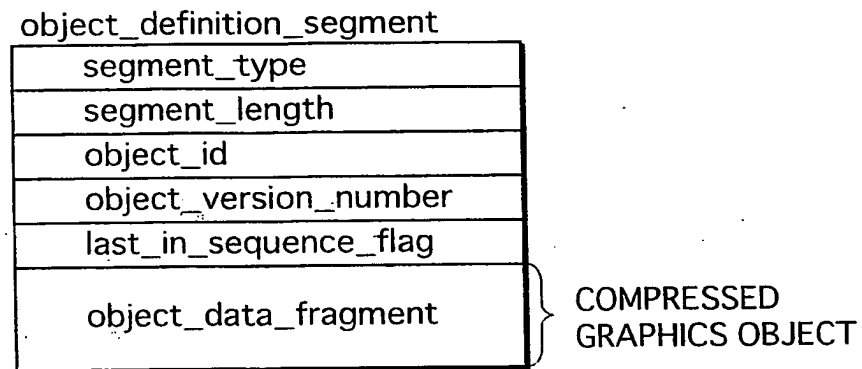
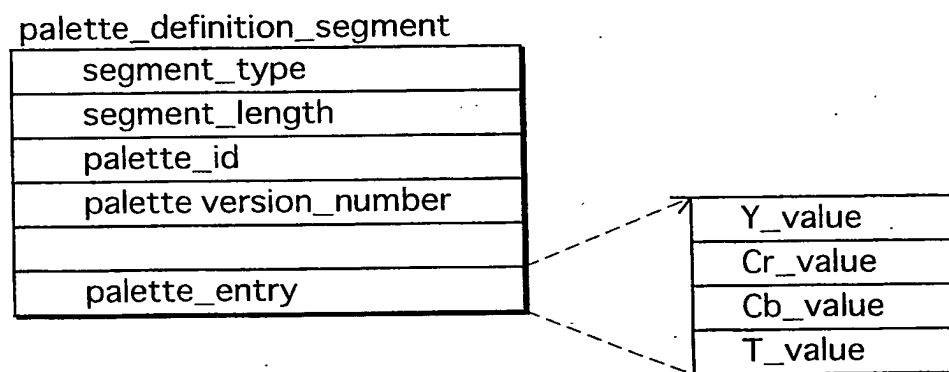


FIG.7B





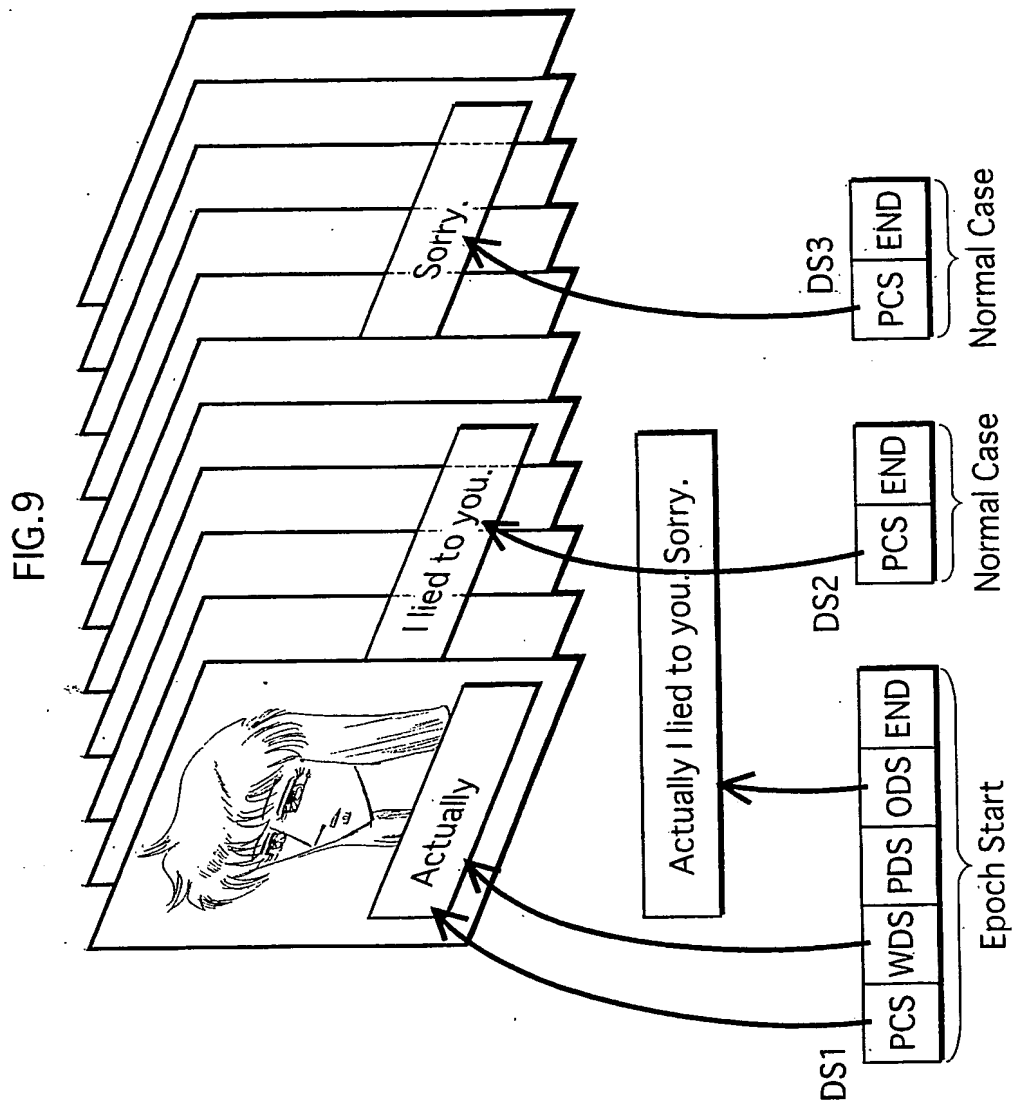


FIG.10

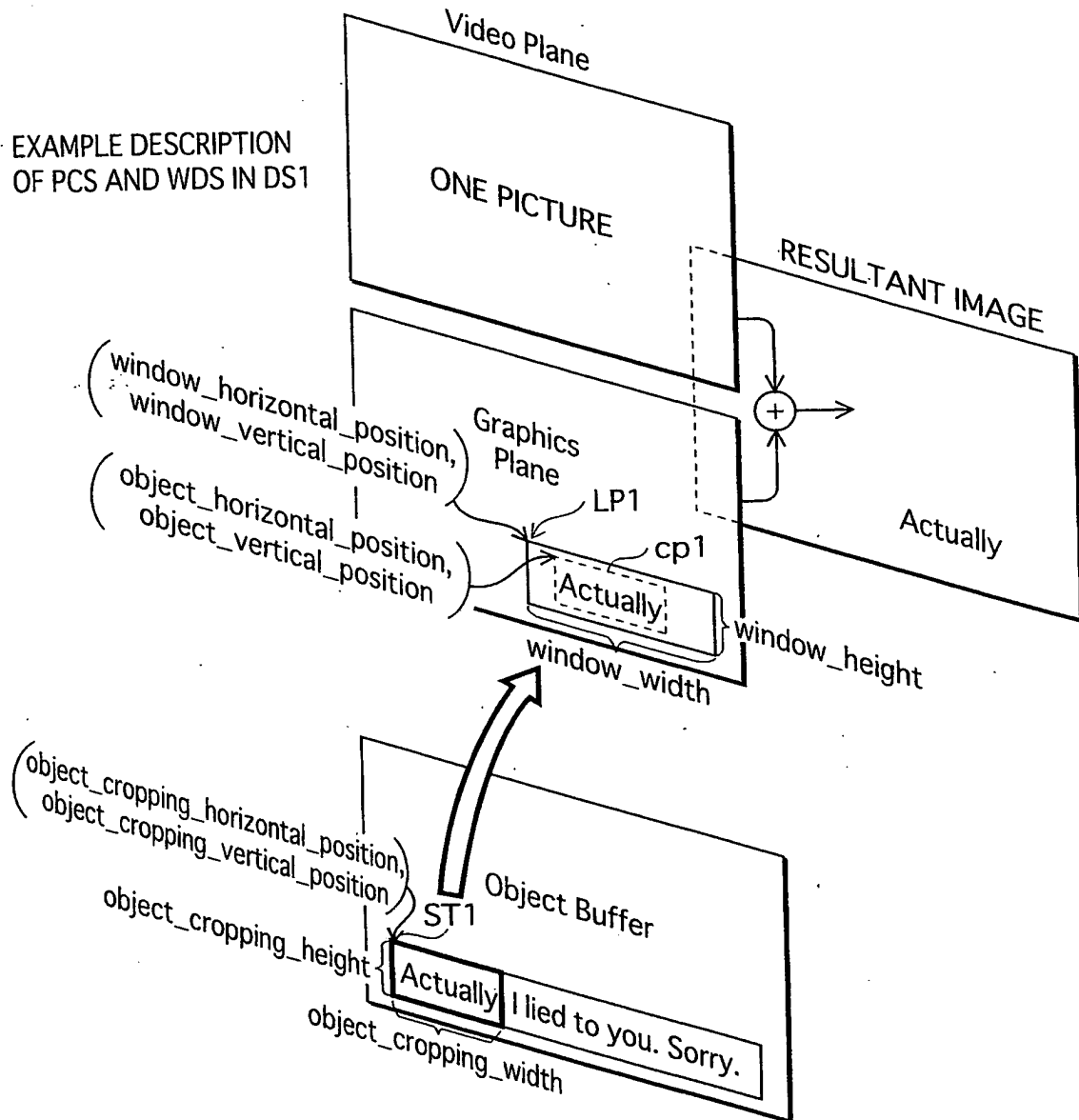
EXAMPLE DESCRIPTION  
OF PCS AND WDS IN DS1

FIG.11

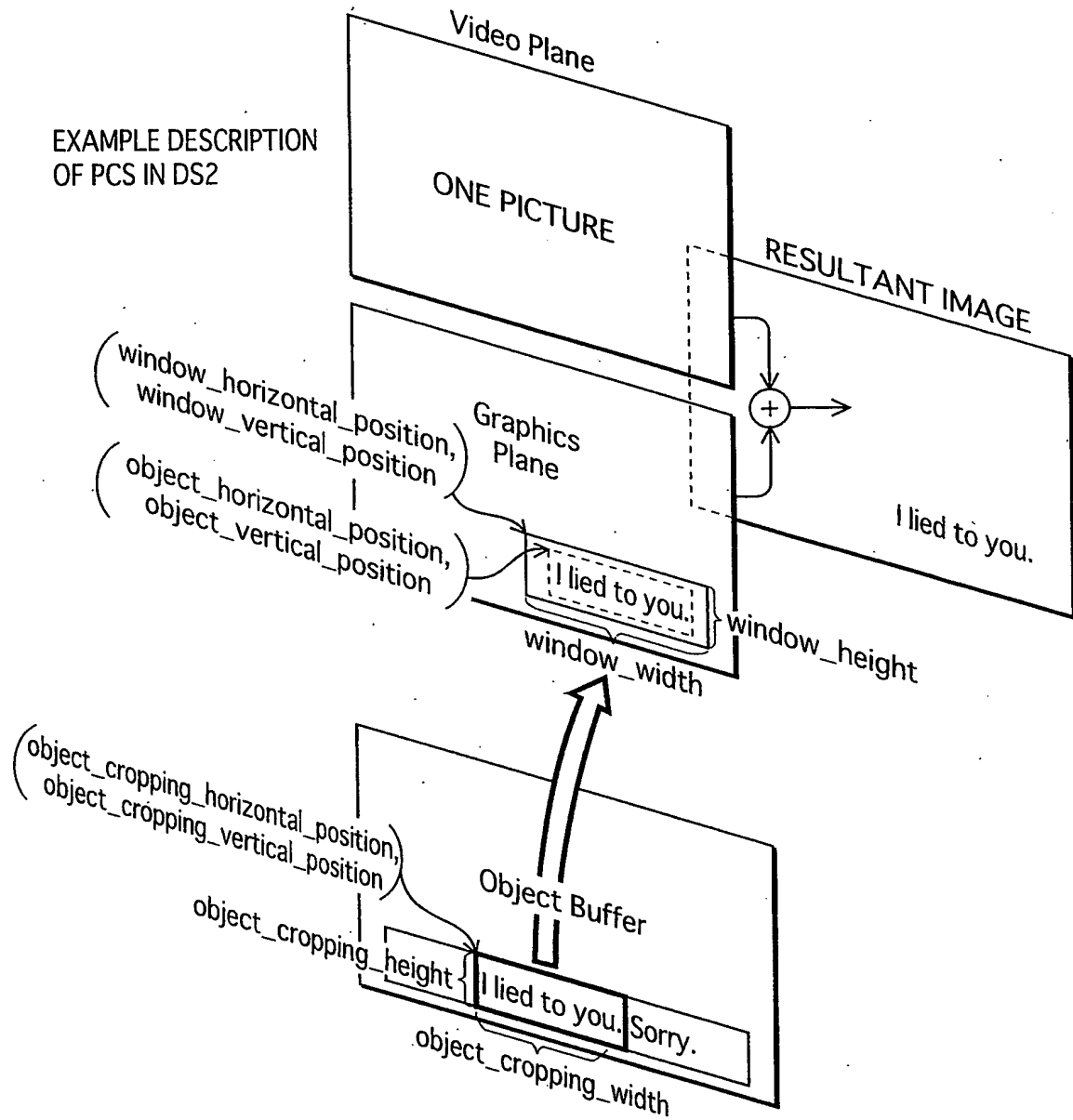
EXAMPLE DESCRIPTION  
OF PCS IN DS2

FIG.12

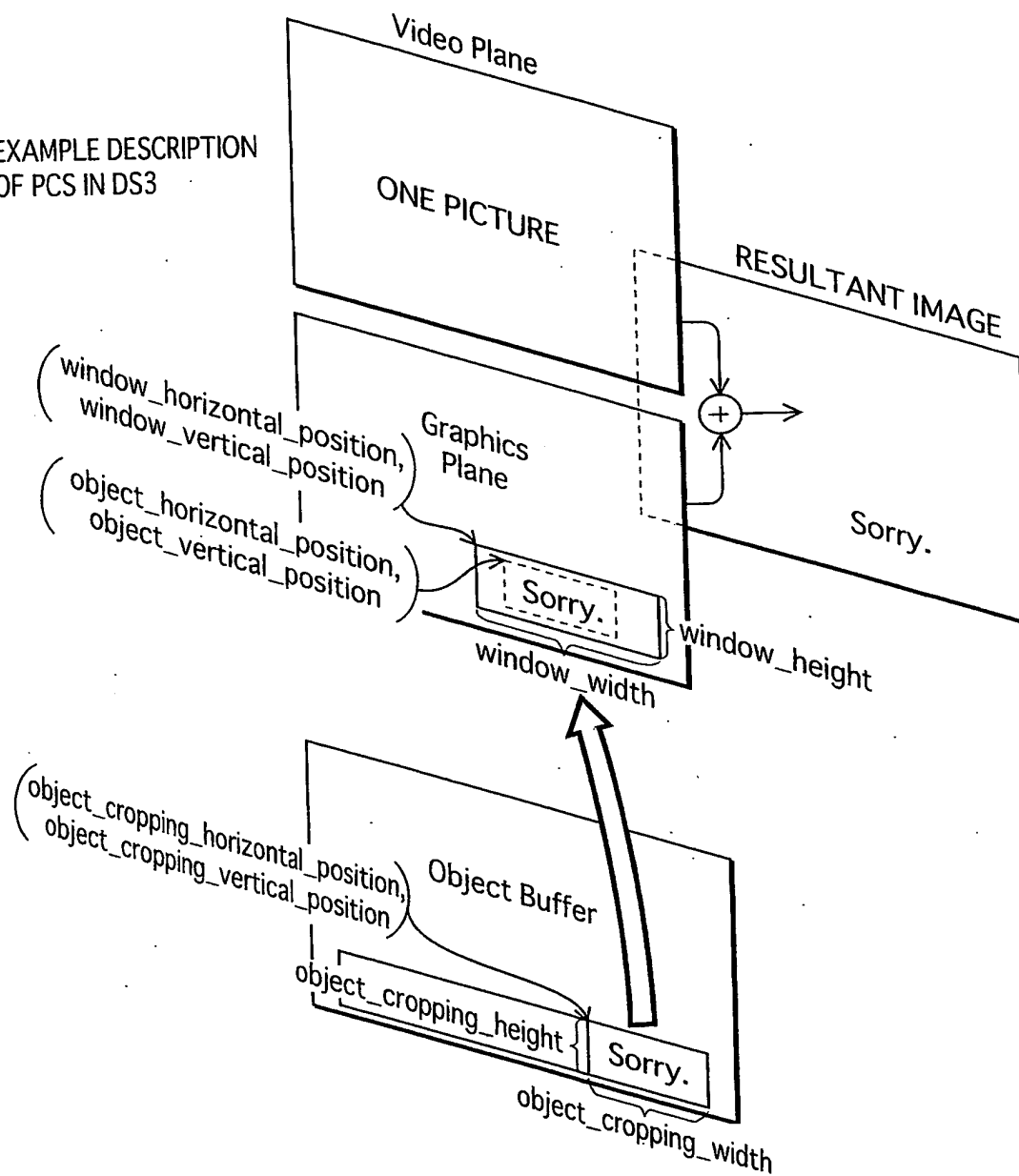
EXAMPLE DESCRIPTION  
OF PCS IN DS3

FIG.13

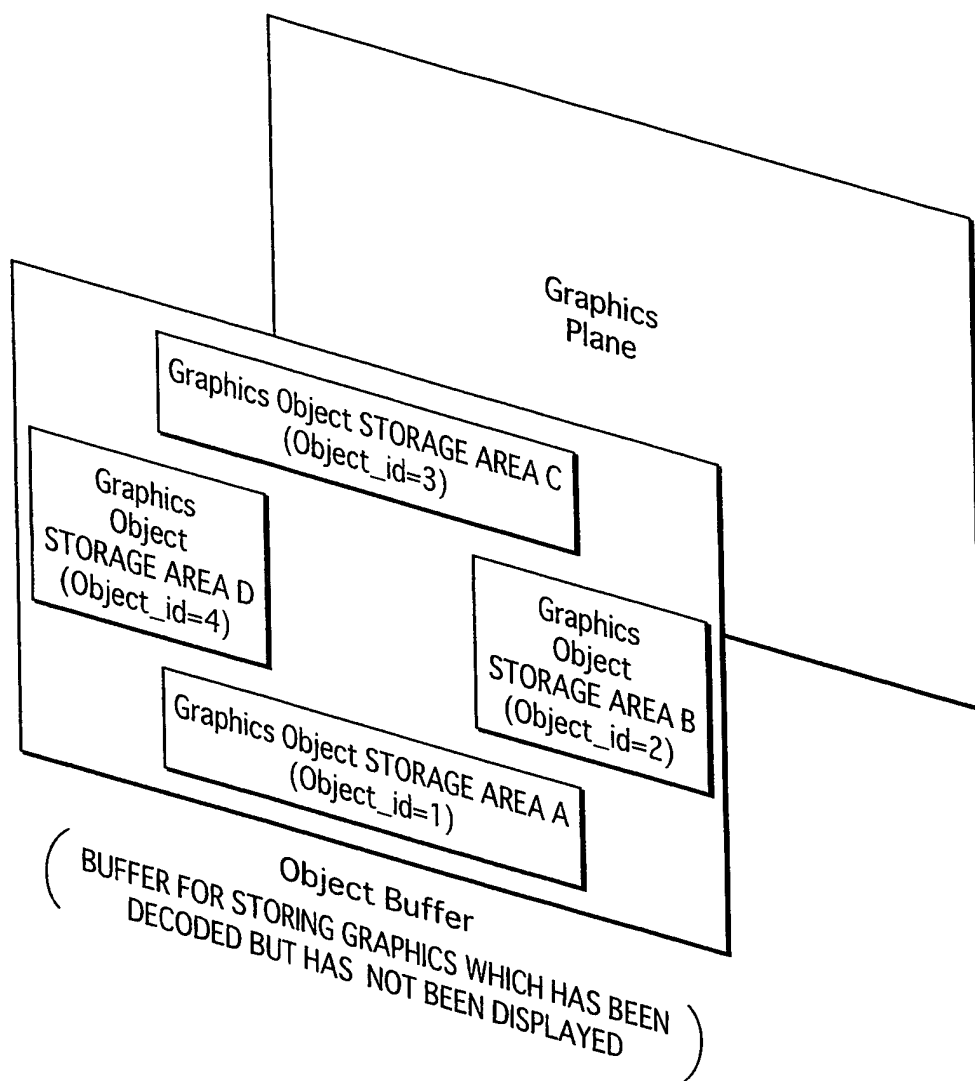


FIG. 14  $PTS( DS_n[PCS]) \geq DTS( DS_n[PCS]) + DECODEDURATION( DS_n)$

Where:

- $DECODEDURATION( DS_n)$  is calculated as follows:

```

decode_duration = 0 ;
decode_duration += PLANEINITIALIZATIONTIME( DS_n ) ;
if( DS_n.PCS.num_of_objects == 2 )
{
    decode_duration += WAIT( DS_n, DS_n.PCS.OBJ[0], decode_duration ) ;
    if( DS_n.PCS.OBJ[0].window_id == DS_n.PCS.OBJ[1].window_id )
    {
        decode_duration += WAIT( DS_n, DS_n.PCS.OBJ[1], decode_duration ) ;
        decode_duration += 90000*( SIZE( DS_n.PCS.OBJ[0].window_id )//256*106 ) ;
    }
    else
    {
        decode_duration += 90000*( SIZE( DS_n.PCS.OBJ[0].window_id )//256*106 ) ;
        decode_duration += WAIT( DS_n, DS_n.PCS.OBJ[1], decode_duration ) ;
        decode_duration += 90000*( SIZE( DS_n.PCS.OBJ[1].window_id )//256*106 ) ;
    }
}
else if( DS_n.PCS.num_of_objects == 1 )
{
    decode_duration += WAIT( DS_n, DS_n.PCS.OBJ[0], decode_duration ) ;
    decode_duration += 90000*( SIZE( DS_n.PCS.OBJ[0].window_id )//256*106 ) ;
}
return decode_duration ;

```

- $PLANEINITIALIZATIONTIME( DS_n)$  is calculated as follows:

```

initialize_duration=0 ;
if( DS_n.PCS.composition_state == EPOCH_START )
{
    initialize_duration = 90000*( 8*video_width*video_height//256*106 ) ;
}
else
{
    for( i=0 ; i < WDS.num_windows ; i++ )
    {
        if( EMPTY( DS_n.WDS.WIN[i], DS_n ) )
            initialize_duration += 90000*( SIZE( DS_n.WDS.WIN[i] )//256*106 ) ;
    }
}
return initialize_duration ;

```

- $WAIT( DS_n, OBJ, current\_duration)$  is calculated as follows:

```

wait_duration = 0 ;
if( EXISTS( OBJ.object_id, DS_n ) )
{
    object_definition_ready_time = PTS( GET( OBJ.object_id, DS_n ) ) ;
    current_time = DTS( DS_n.PCS ) + current_duration ;
    if( current_time < object_definition_ready_time )
        wait_duration += object_definition_ready_time - current_time ;
}
return wait_duration ;

```

FIG. 15

## CALCULATION OF DECODEDURATION

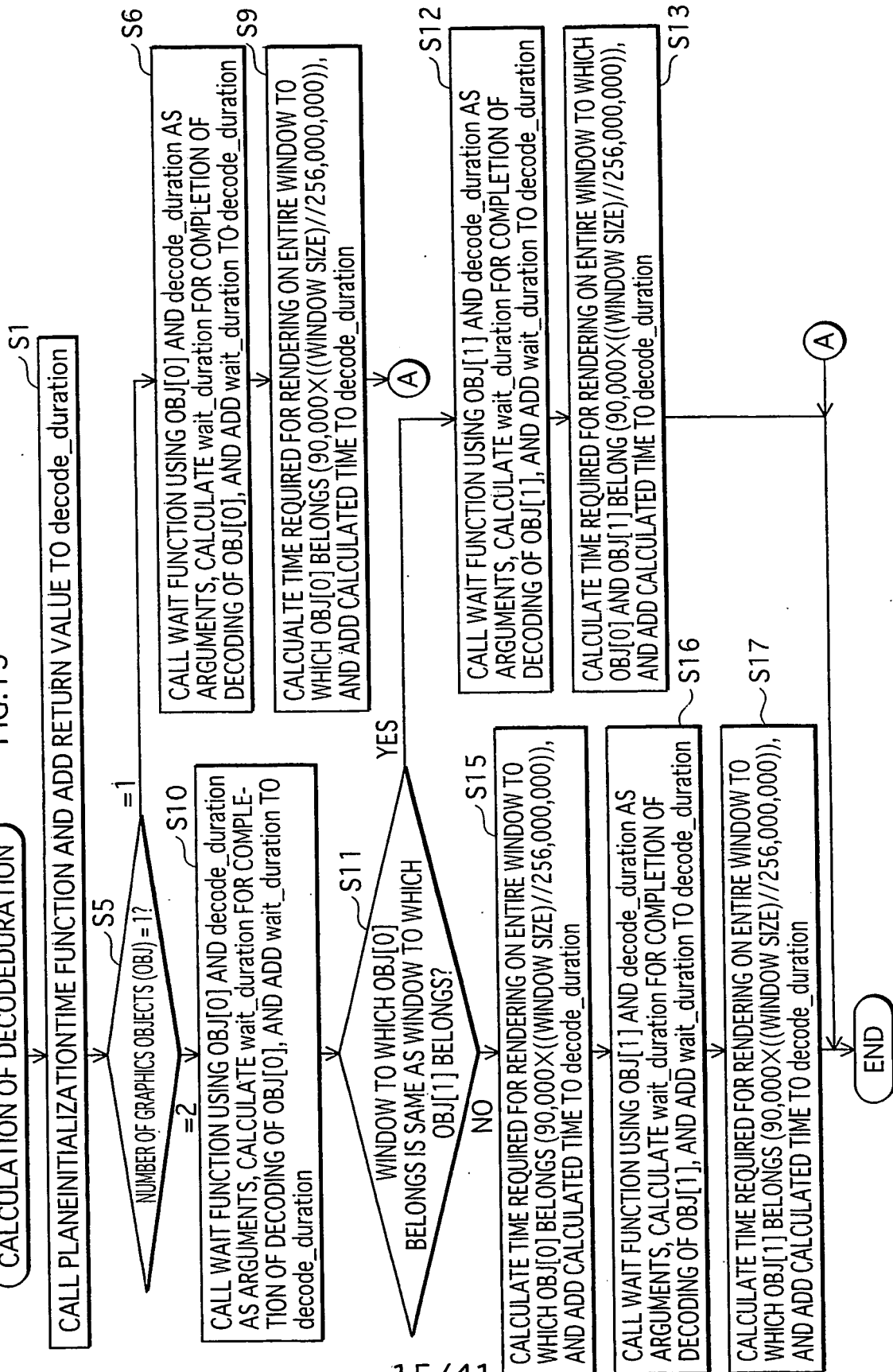


FIG.16A

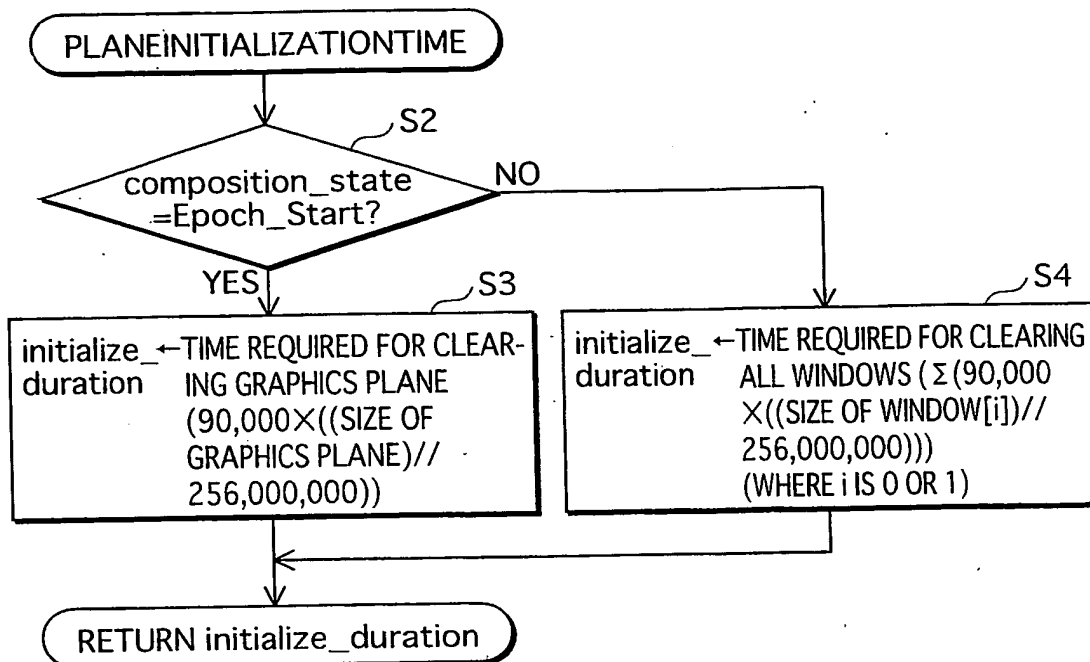


FIG.16B

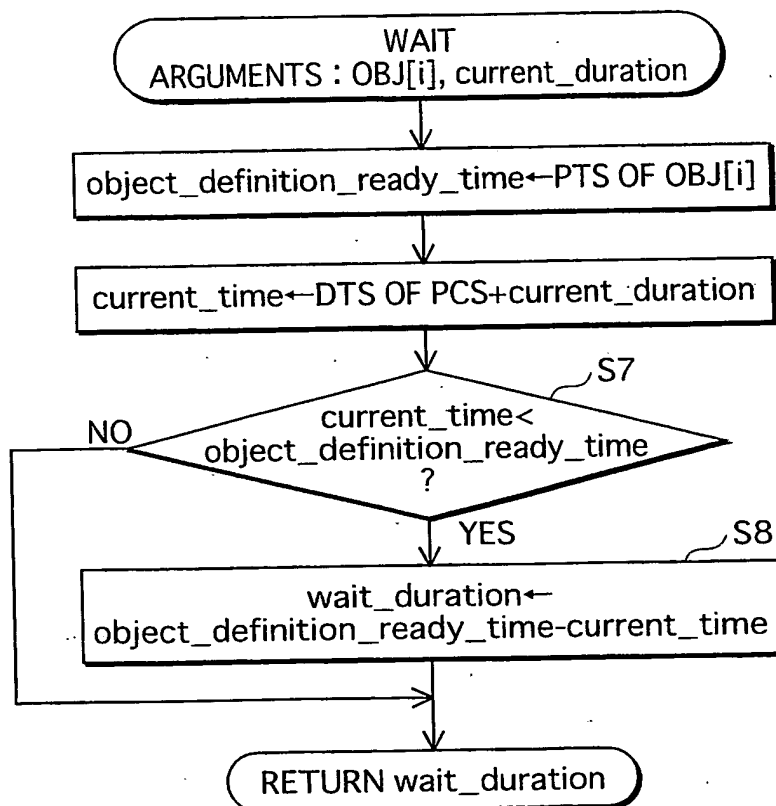


FIG.17A

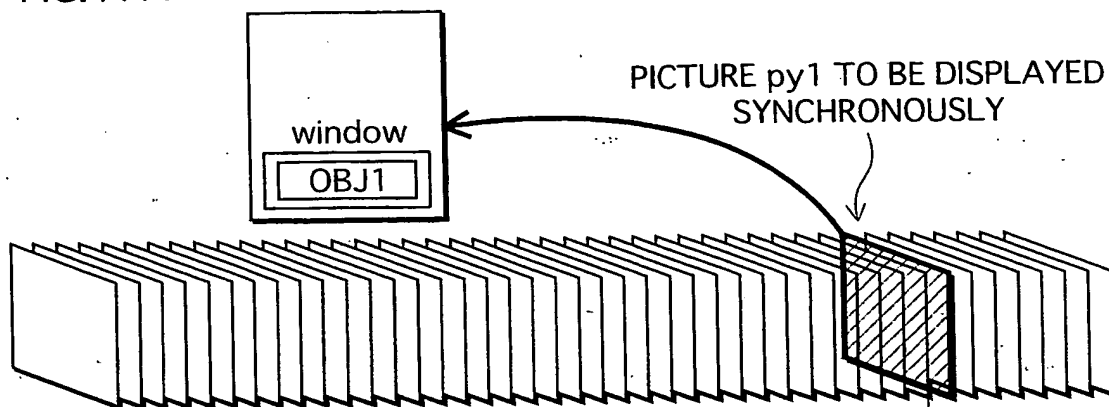


FIG.17B

DECODE\_DURATION  
=(2)+(3)

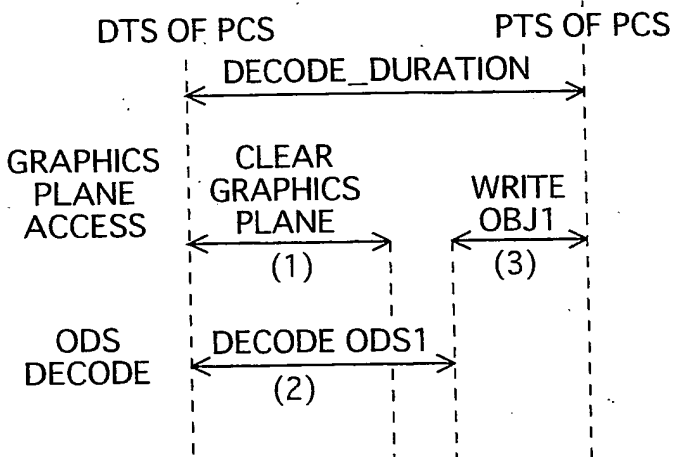


FIG.17C

DECODE\_DURATION  
=(1)+(3)

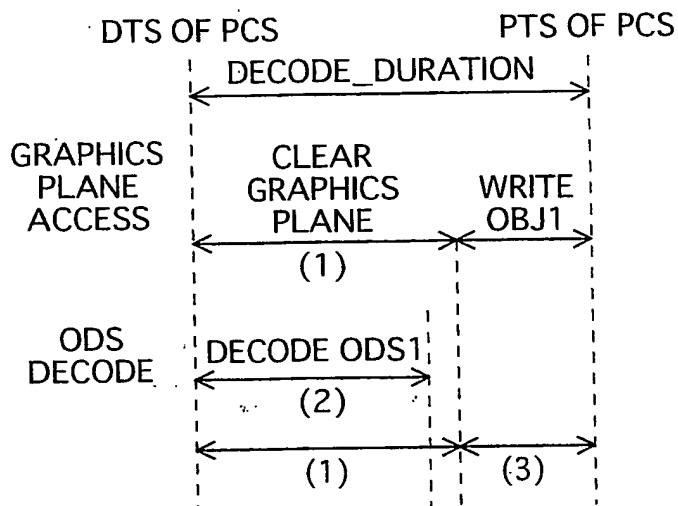


FIG.18A

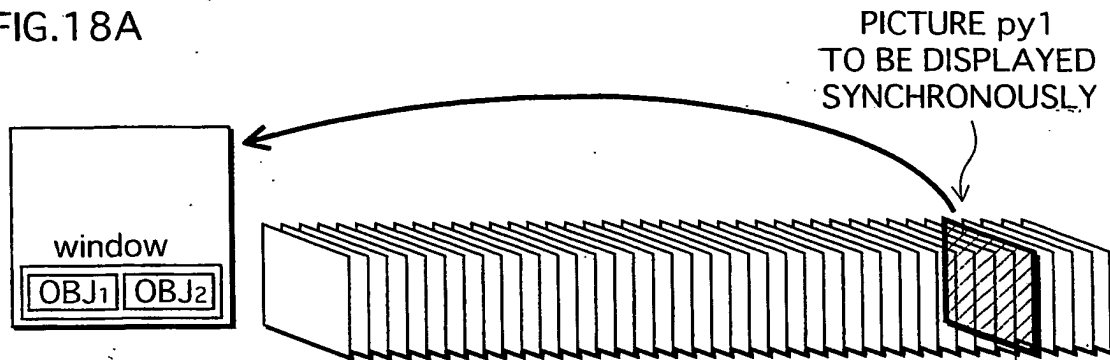


FIG.18B

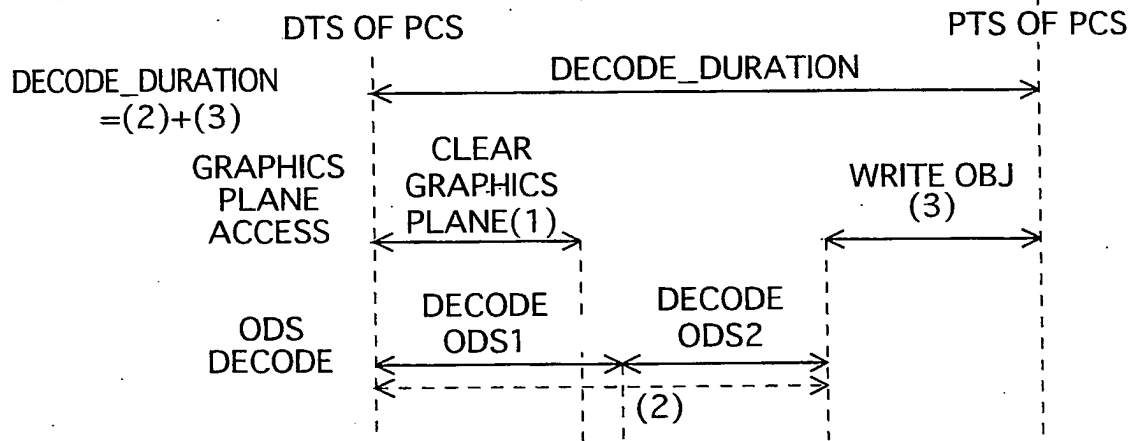


FIG.18C

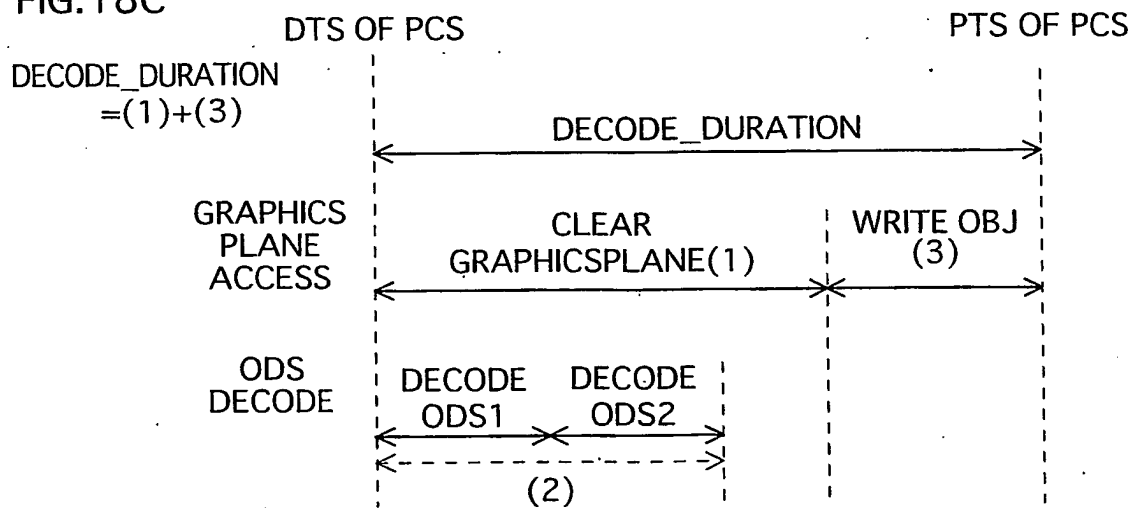




FIG.20

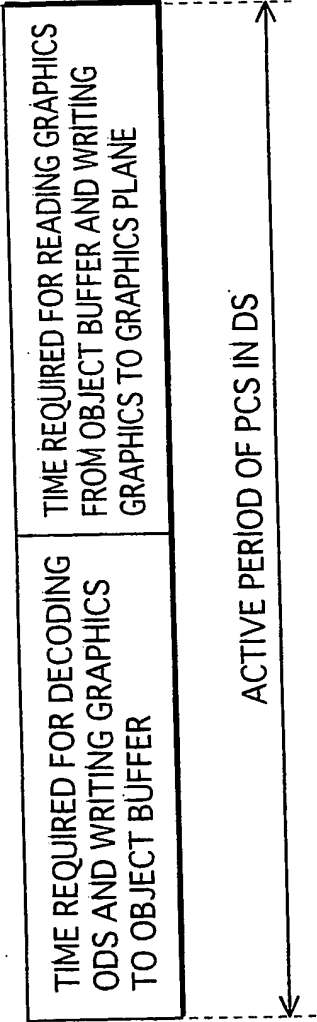


FIG.21

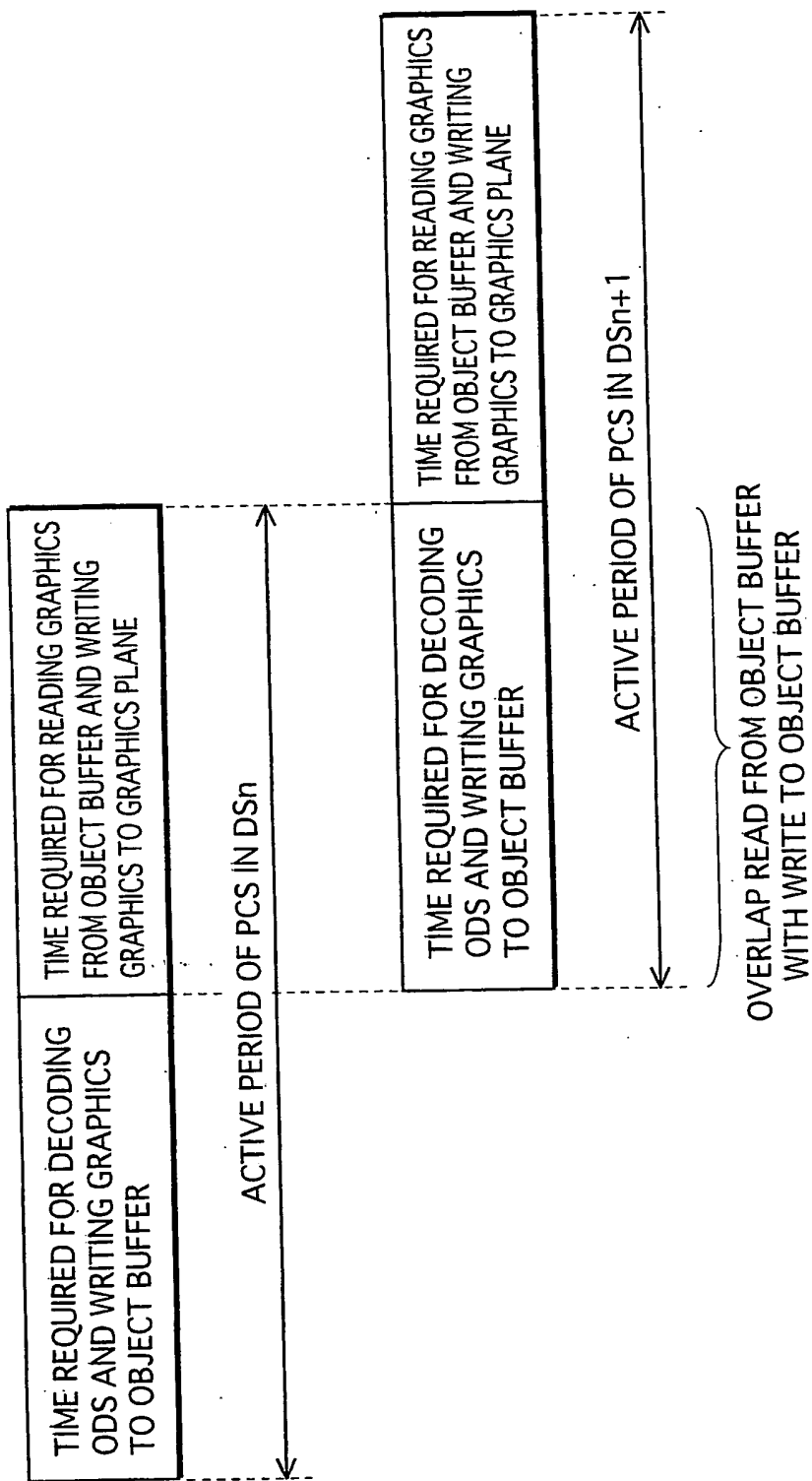


FIG.22

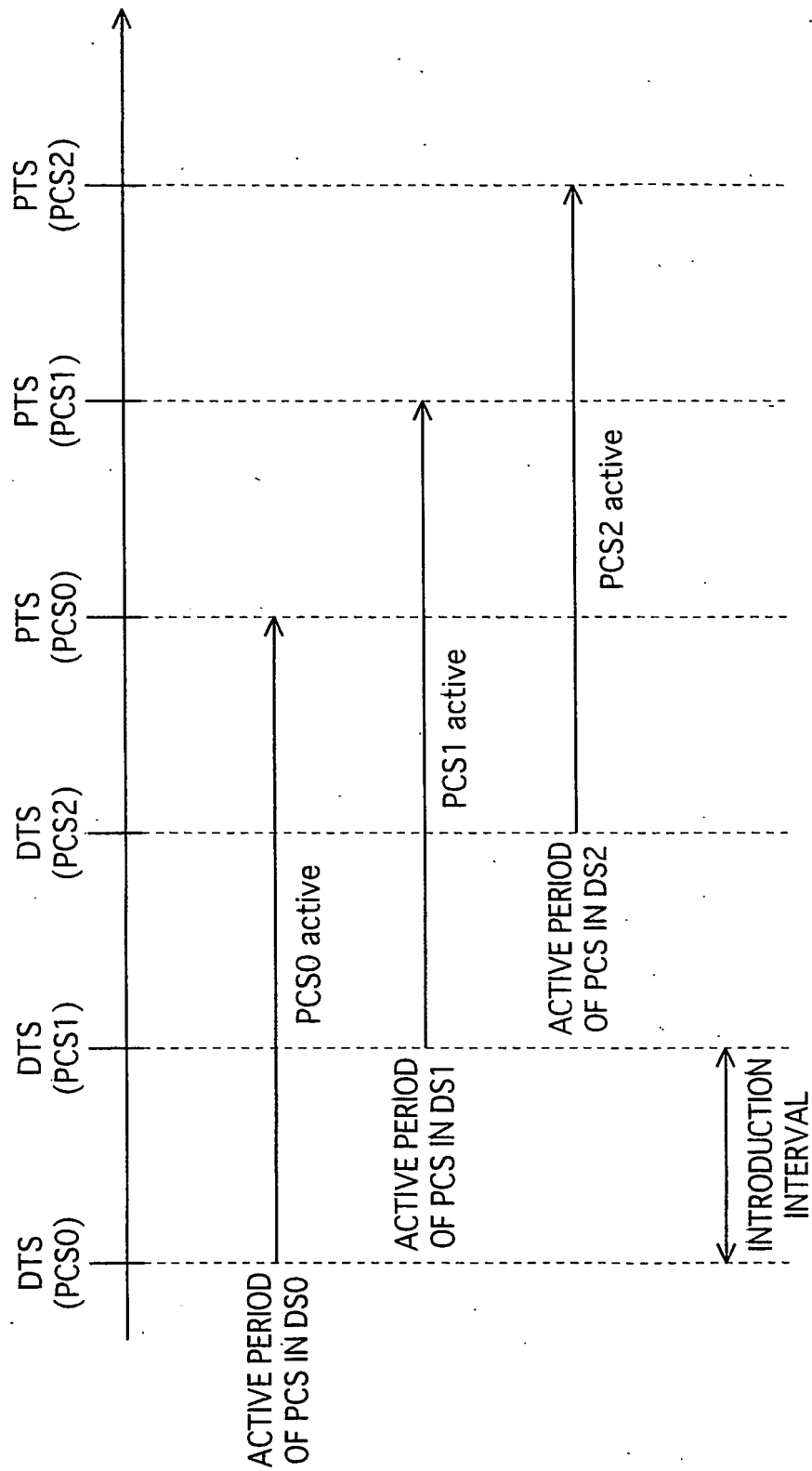


FIG.23

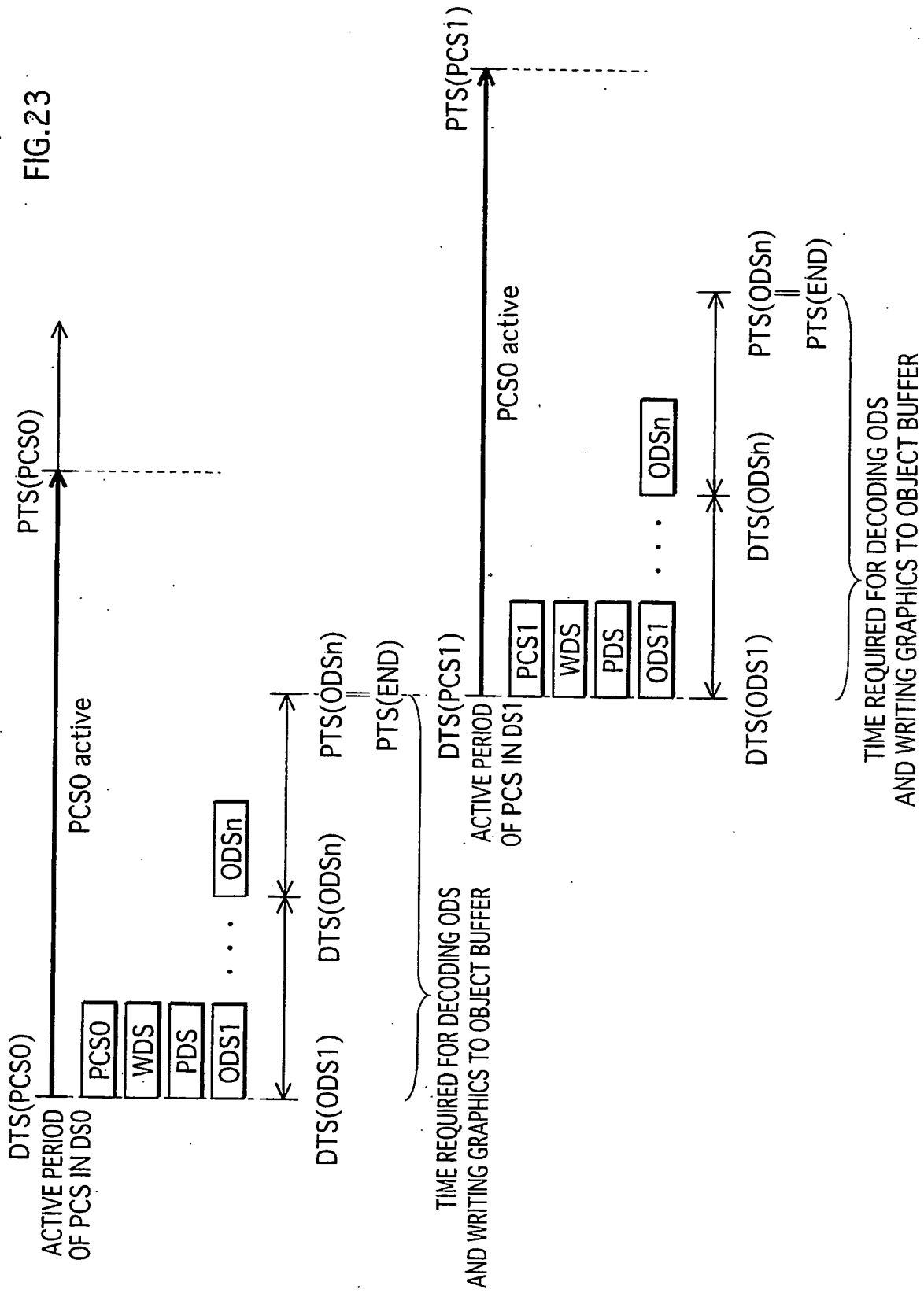


FIG.24

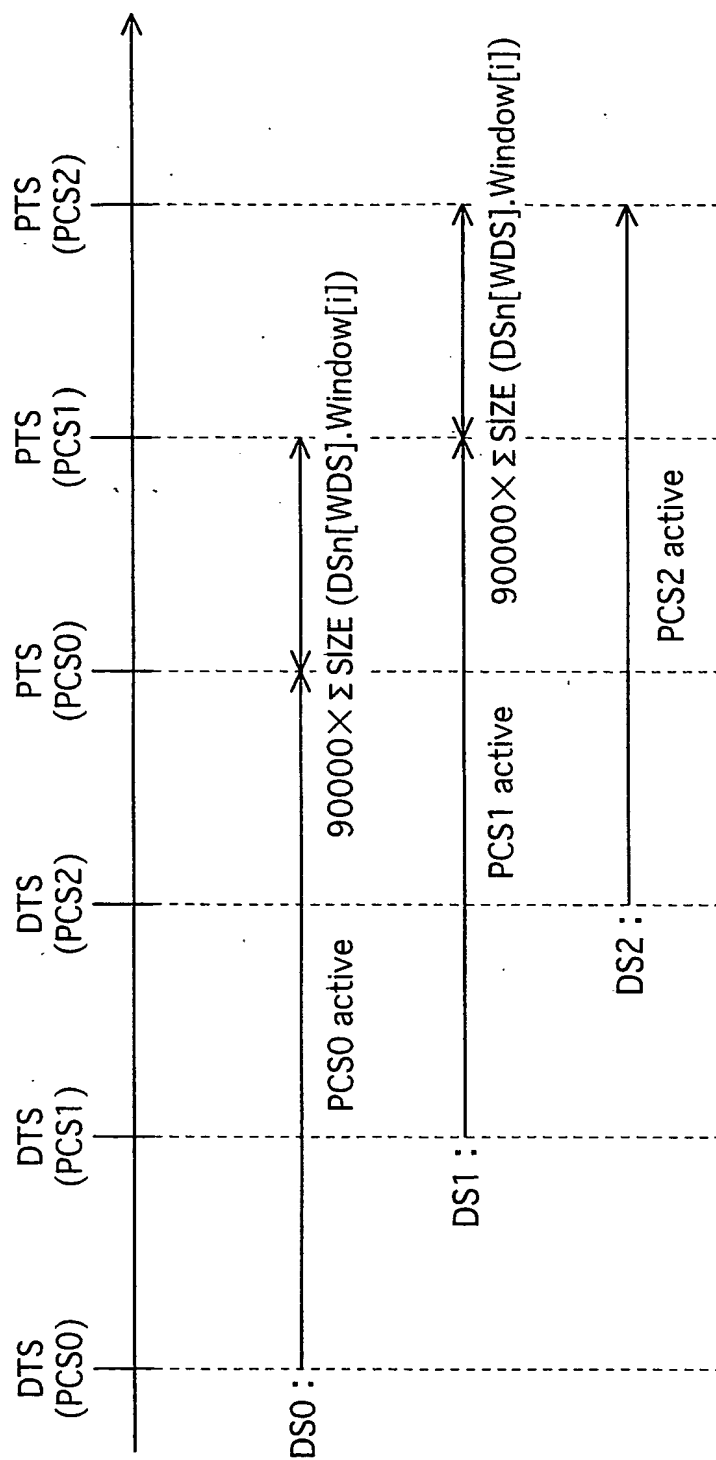


FIG.25A PIPELINE

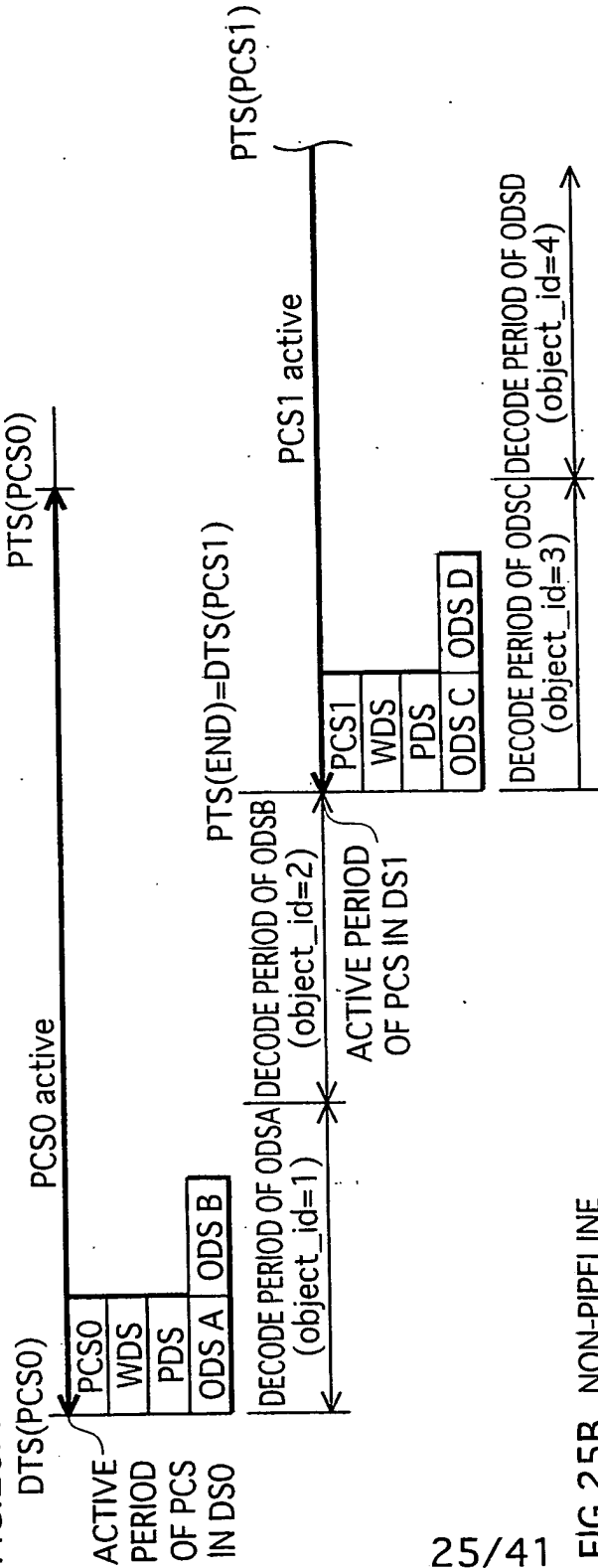


FIG.25B NON-PIPELINE

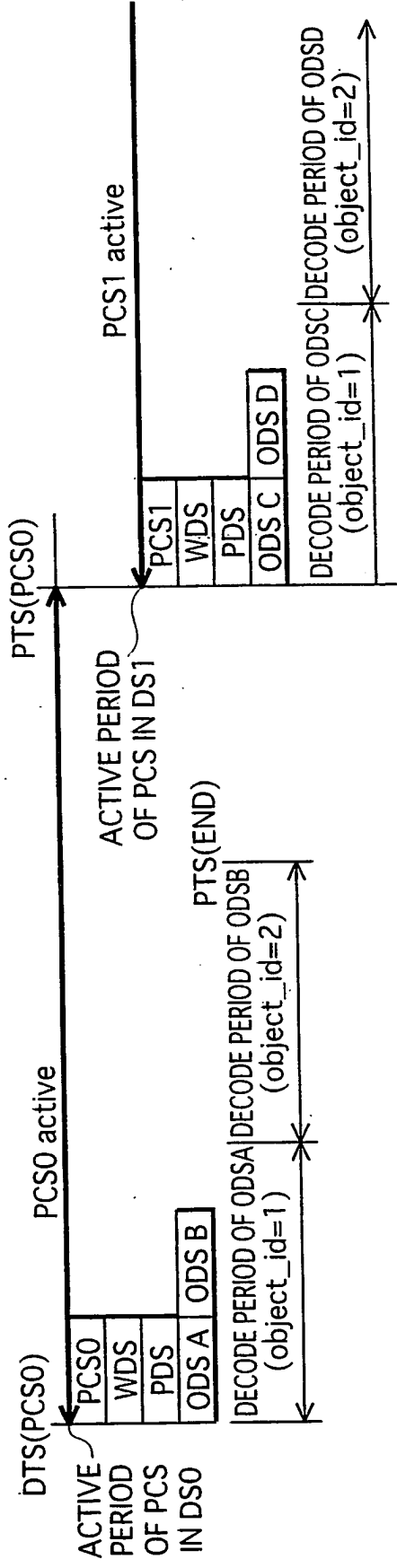


FIG.26

END SEGMENT SHOWS  
END OF TRANSFER OF  
ODSS IN DS

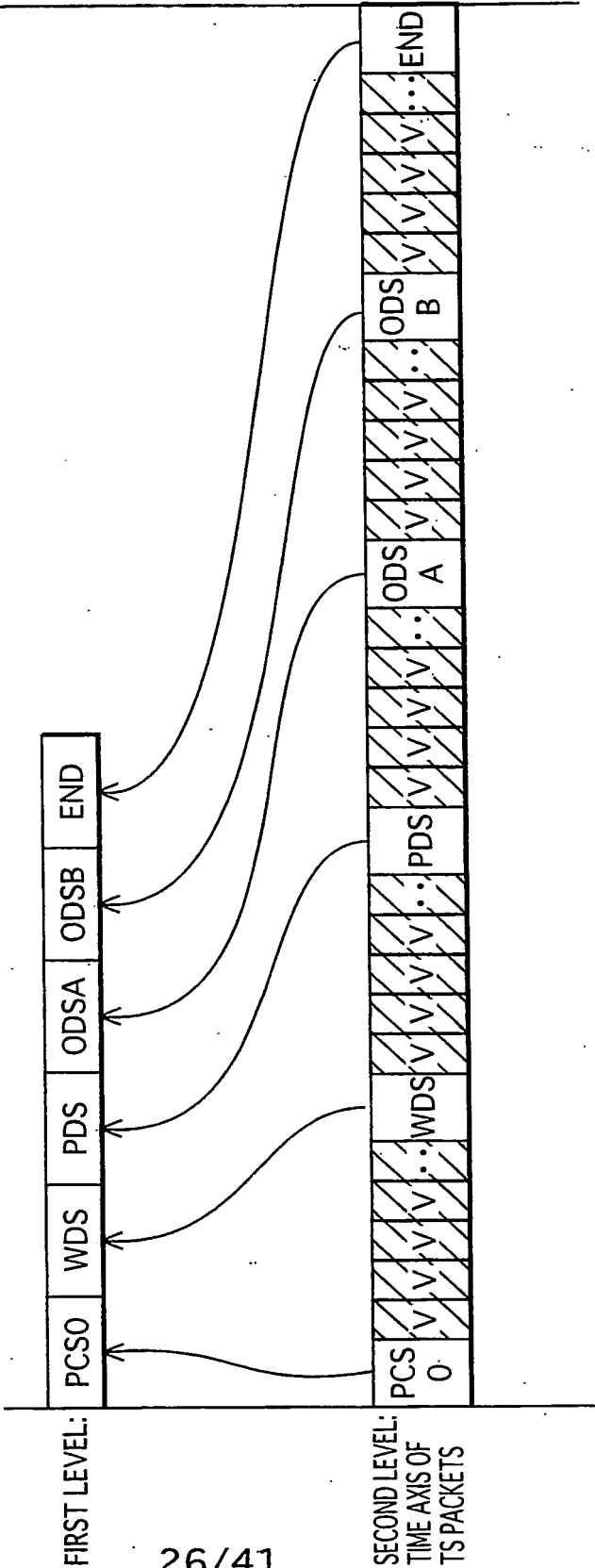


FIG.27A SCREEN COMPOSITION

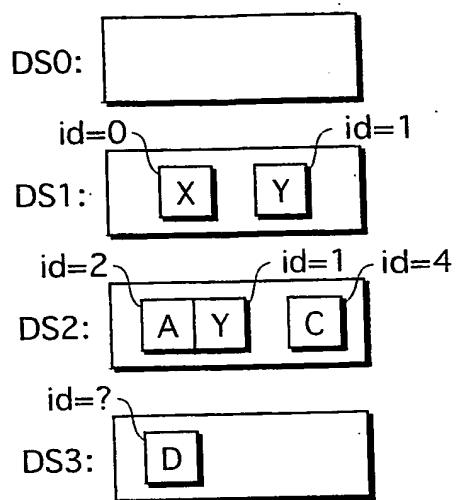


FIG.27B ACTIVE PERIOD OVERLAPPING AND ODS TRANSFER

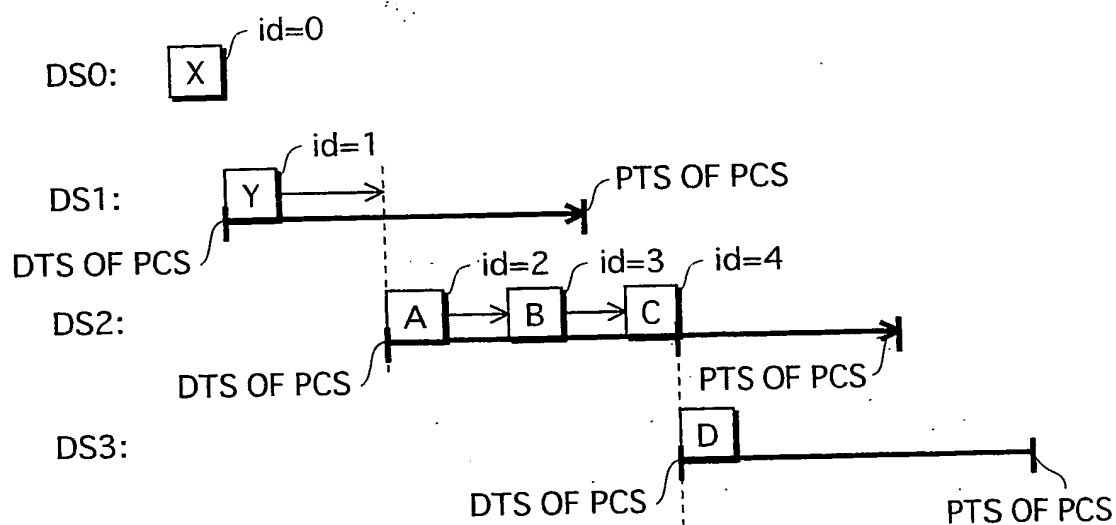


FIG.27C ARRANGEMENT IN OBJECT BUFFER

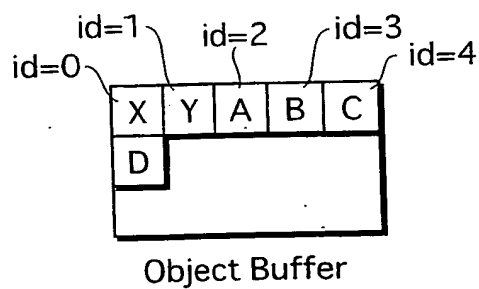


FIG.28

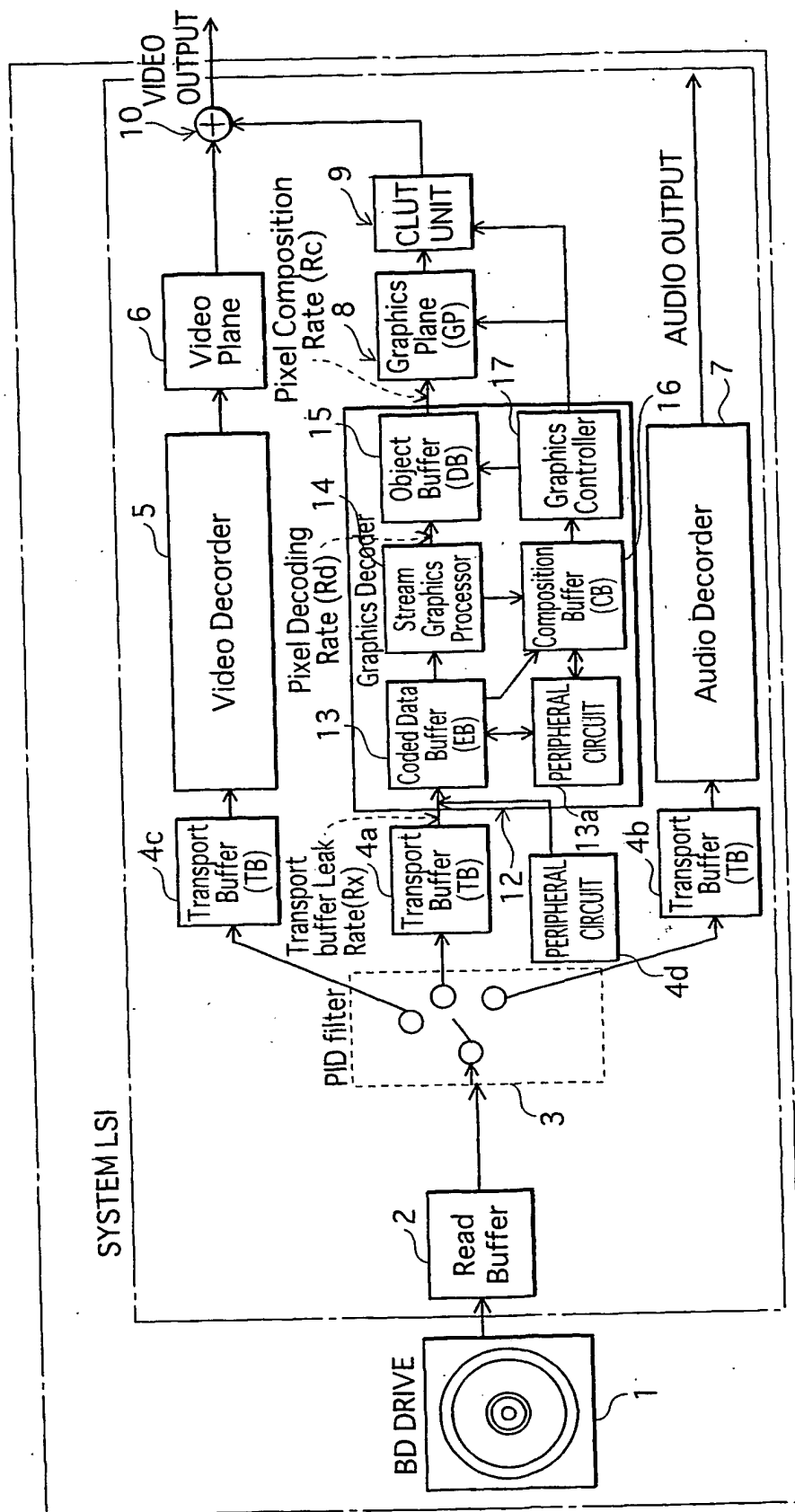
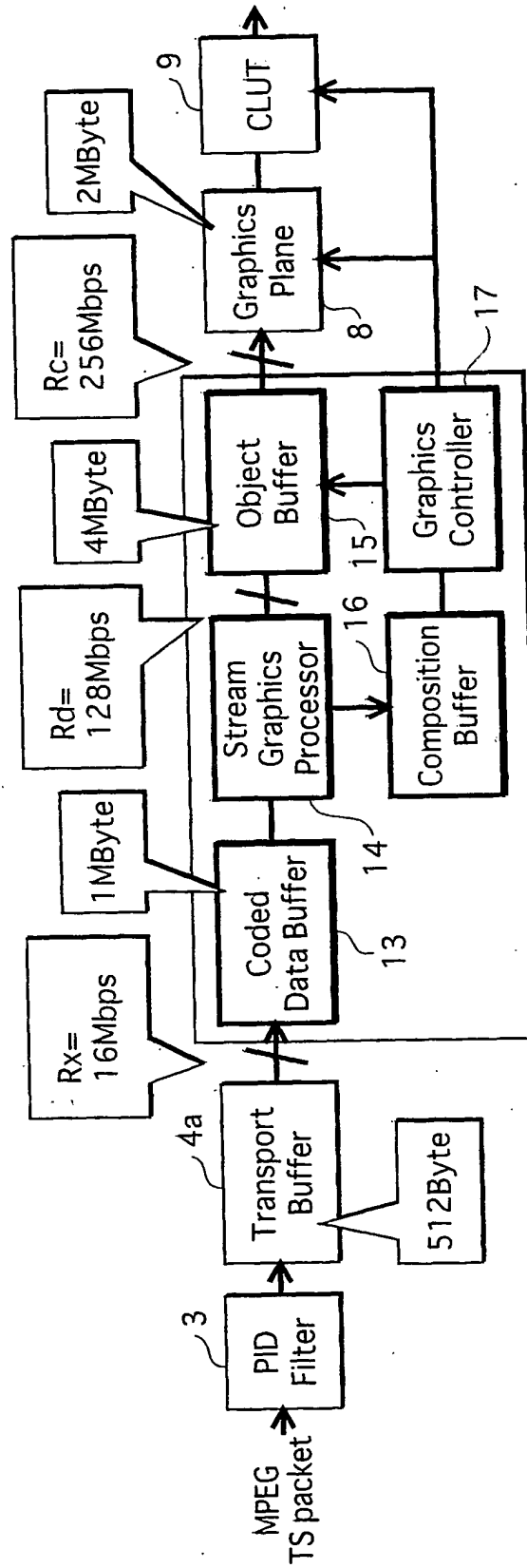
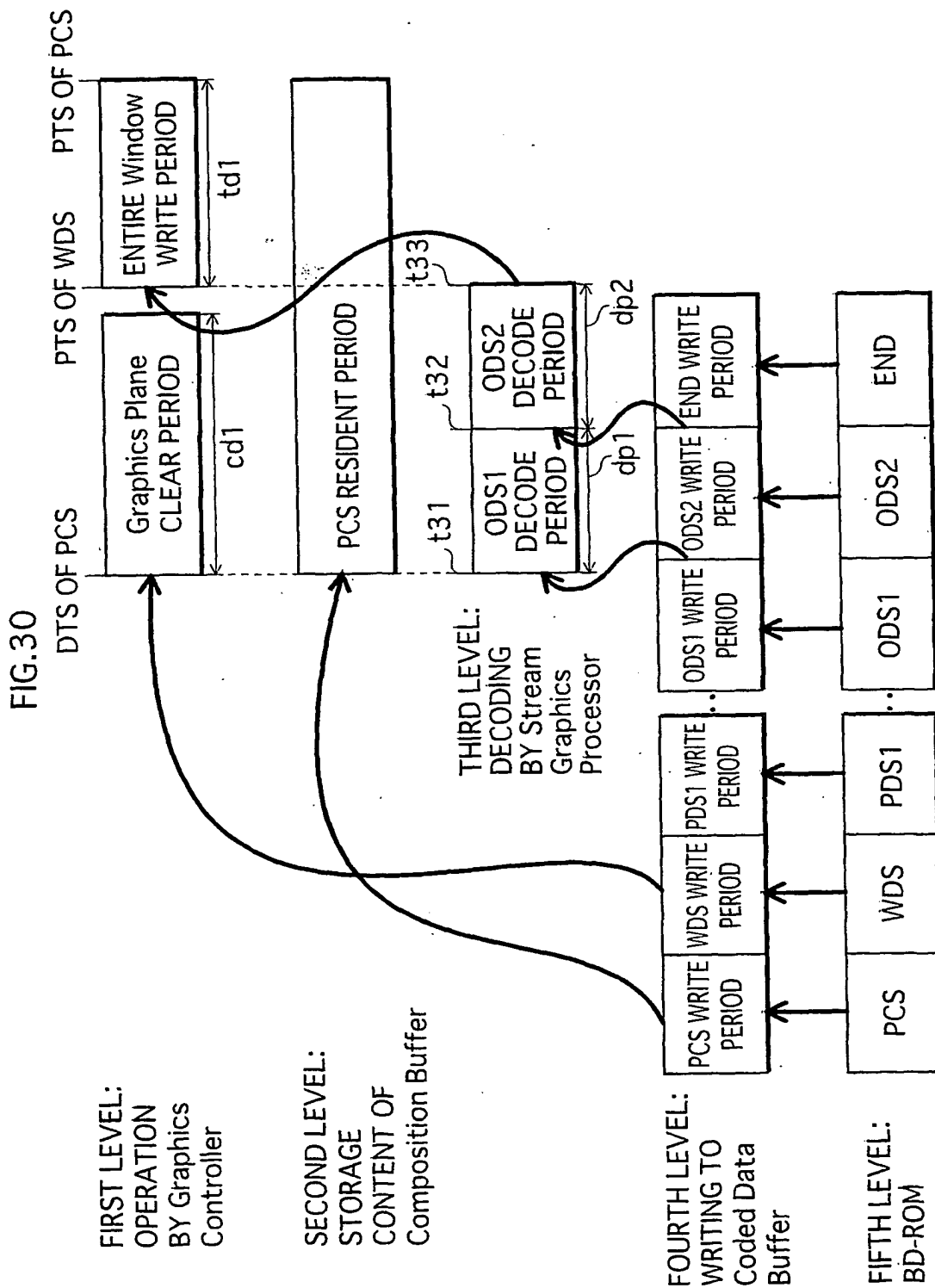
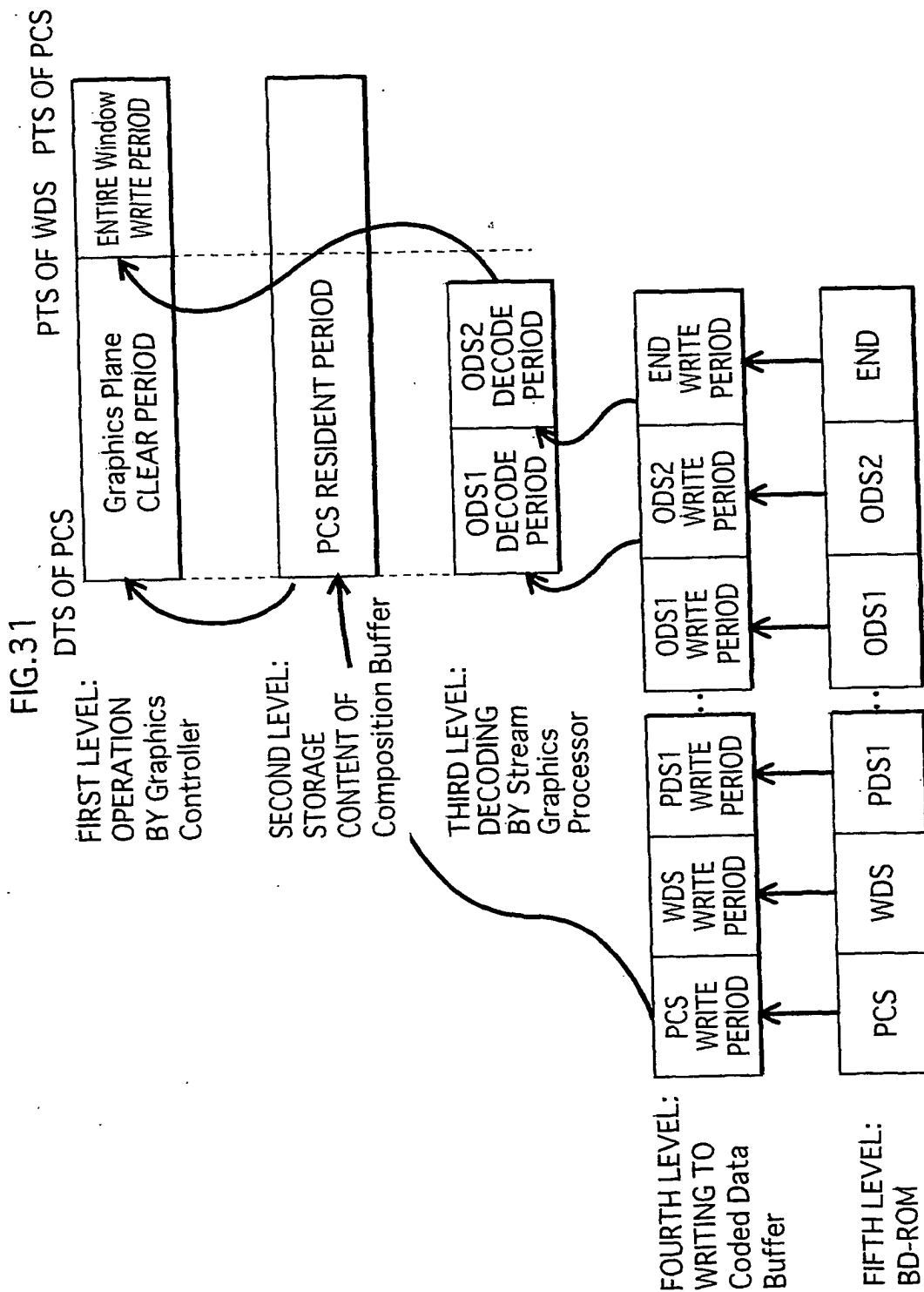


FIG.29







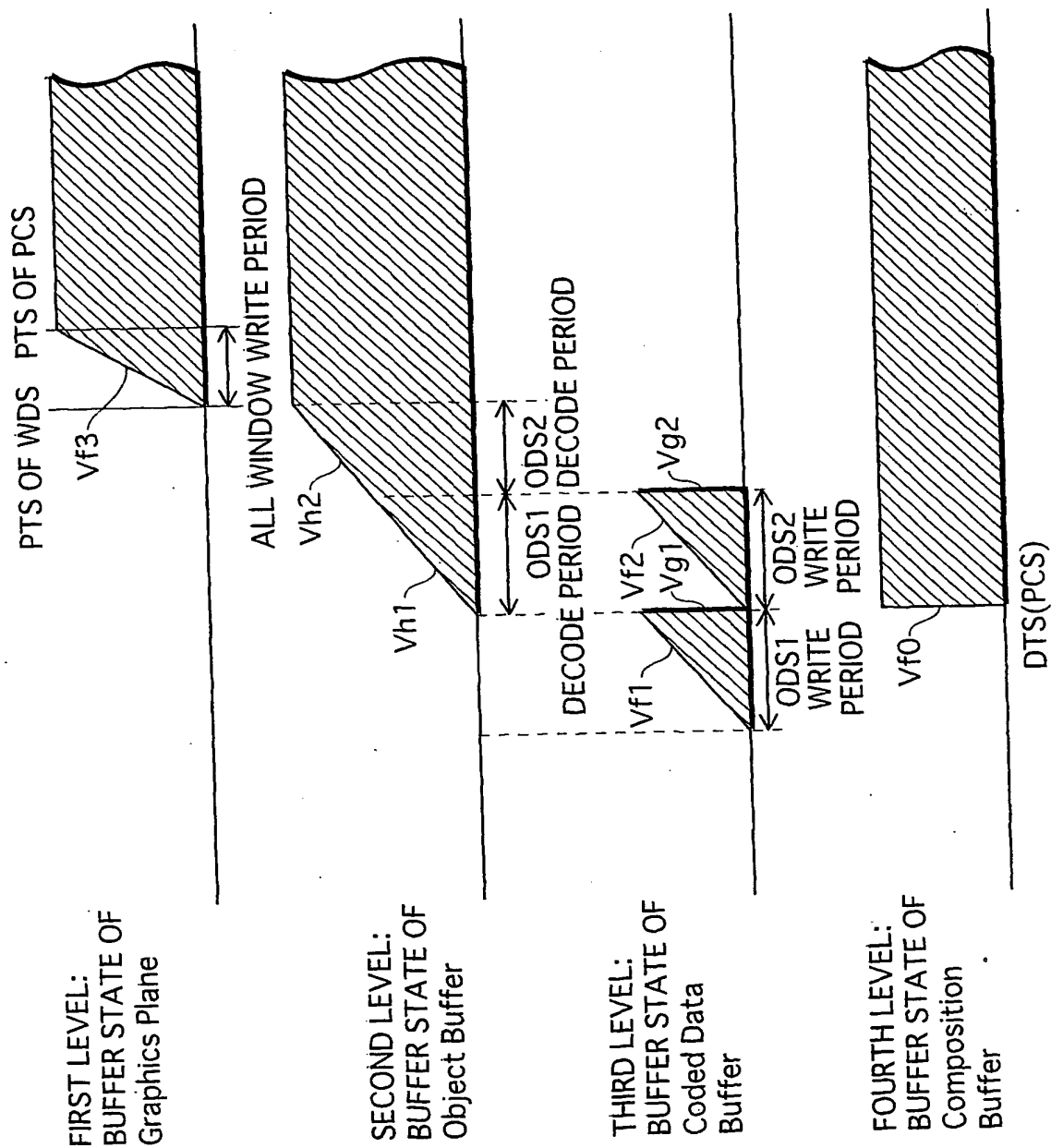


FIG. 32

FIG. 33

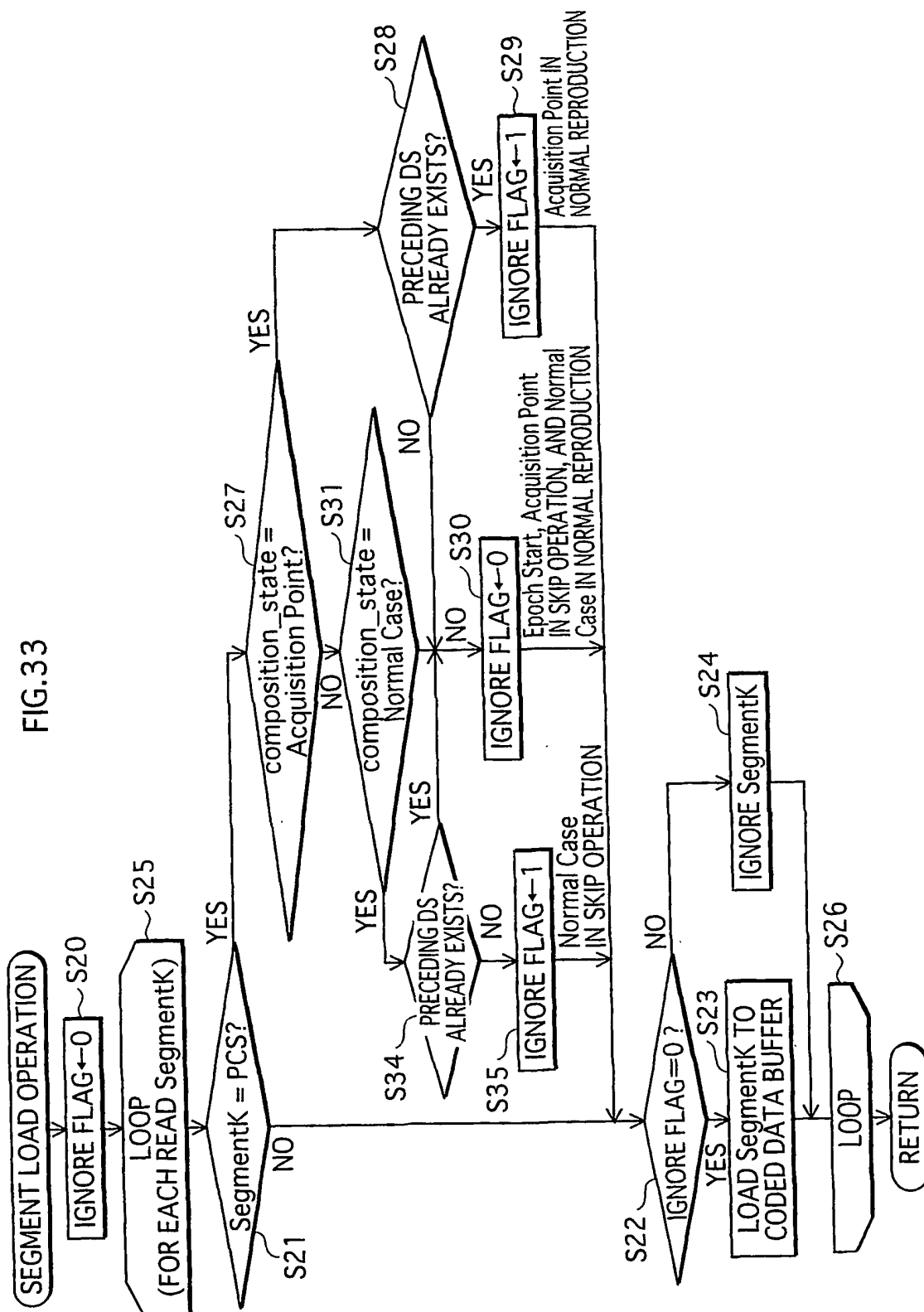


FIG.34

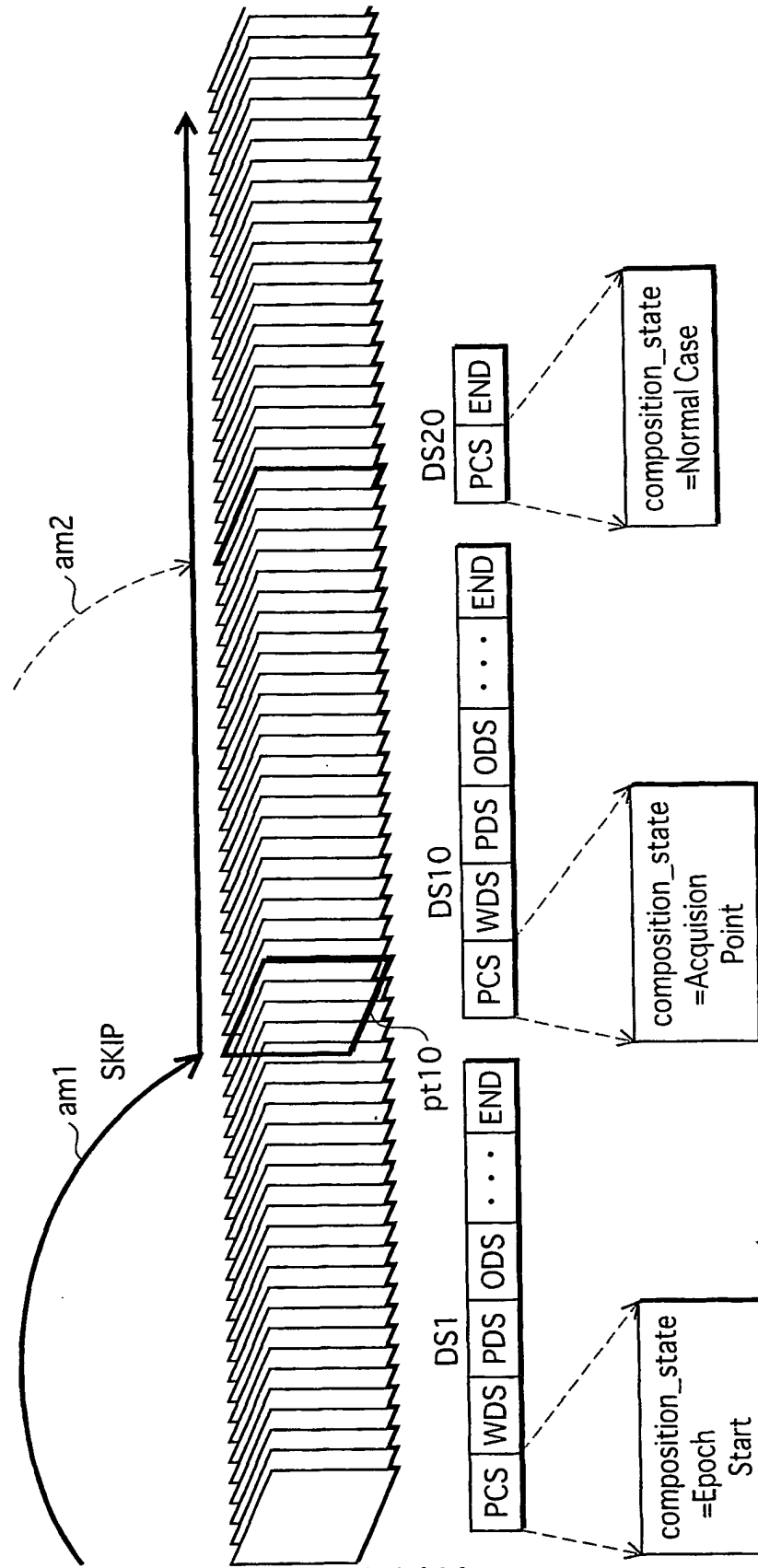


FIG.35  
Coded Data Buffer IN REPRODUCTION APPARATUS

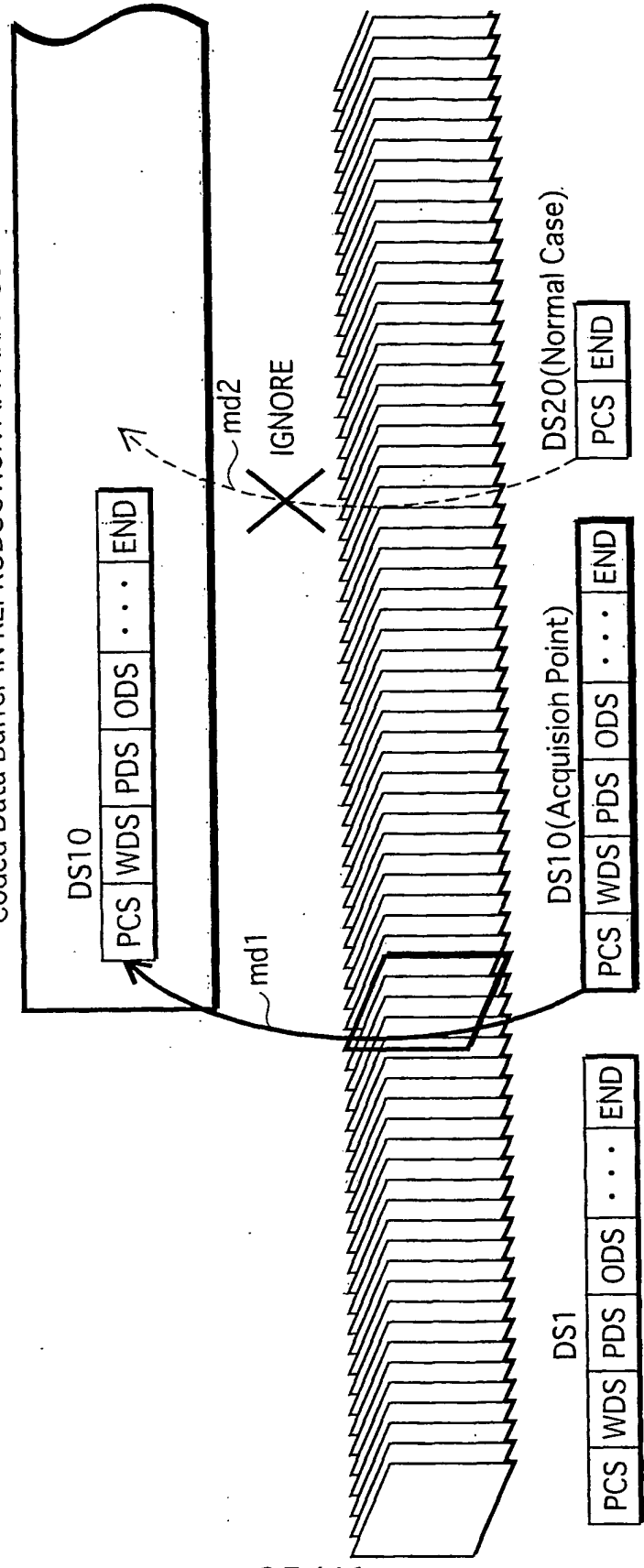


FIG.36

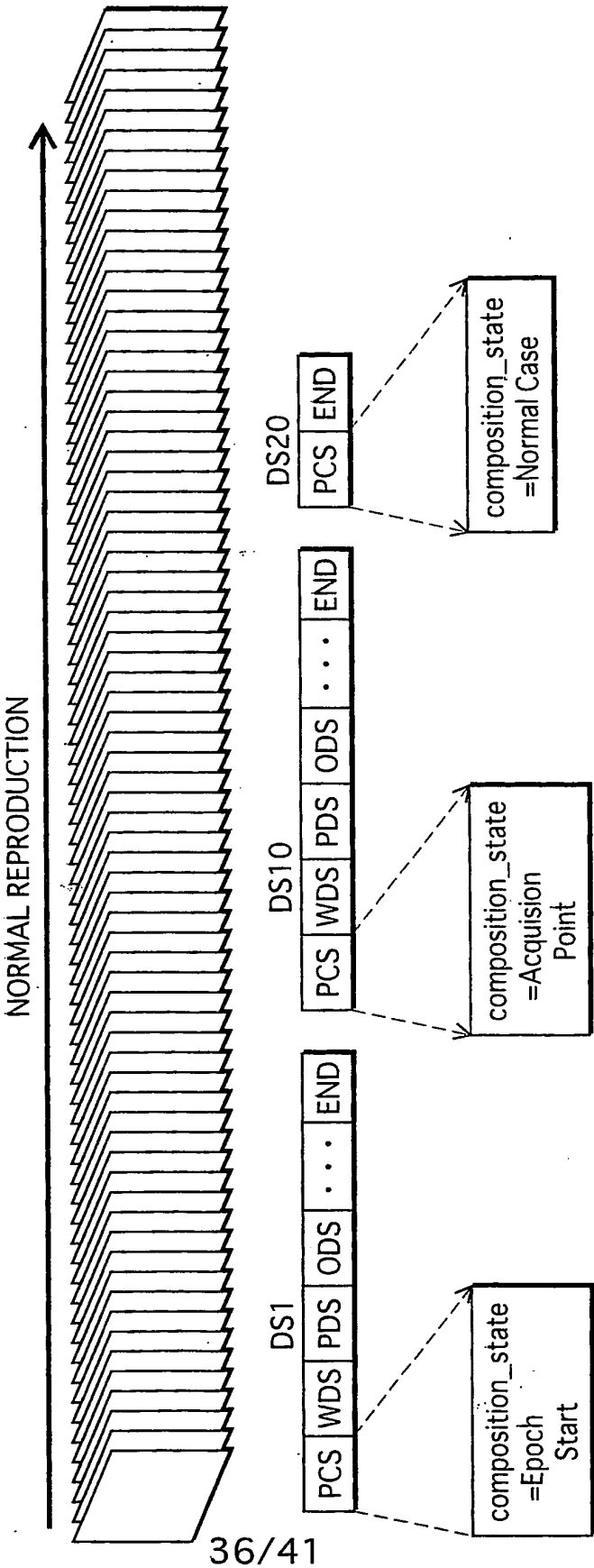


FIG.37

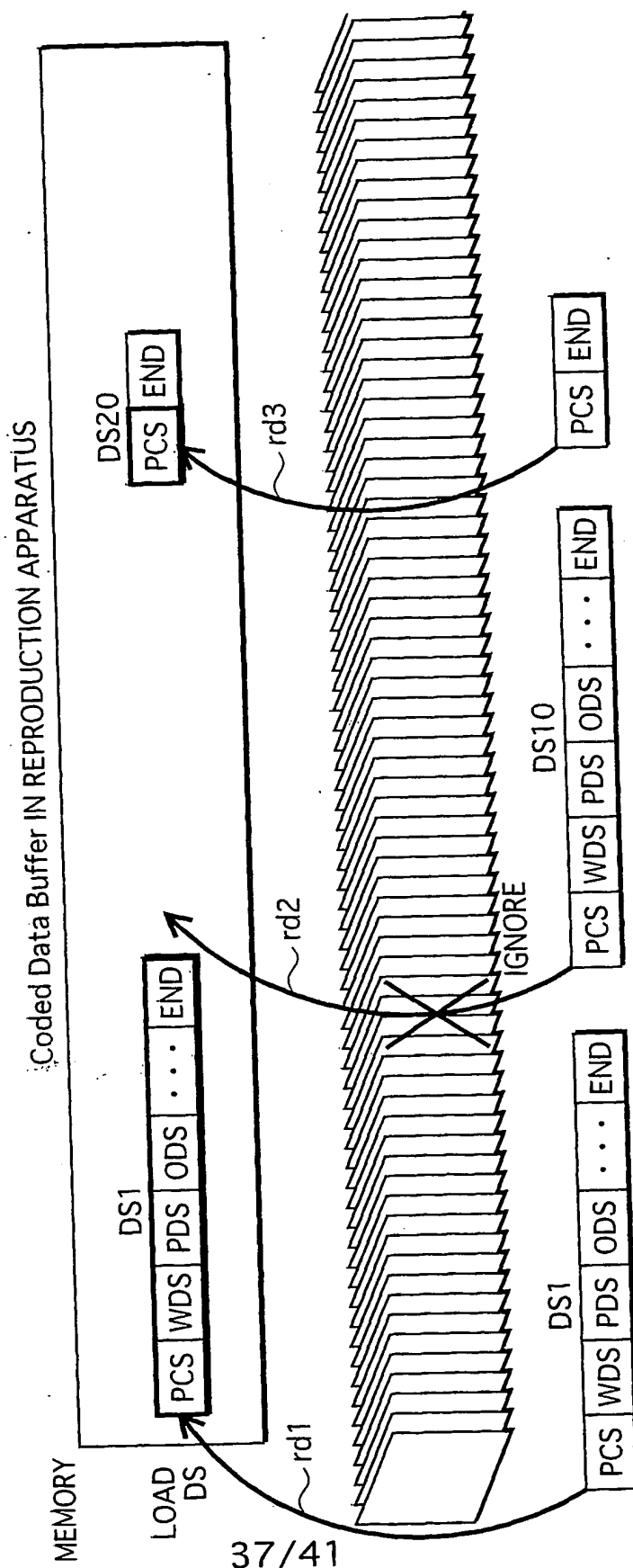




FIG. 39

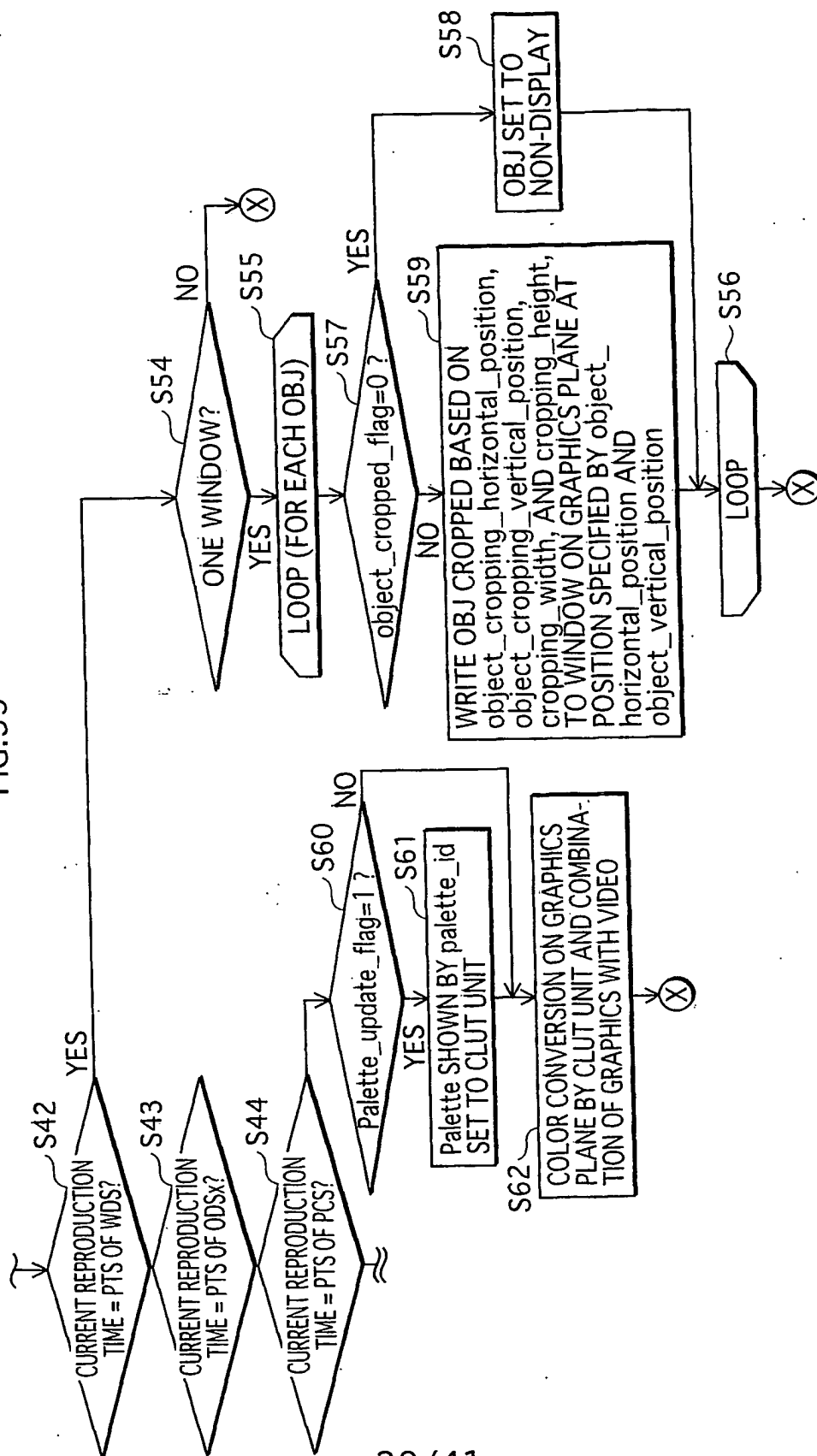


FIG.40

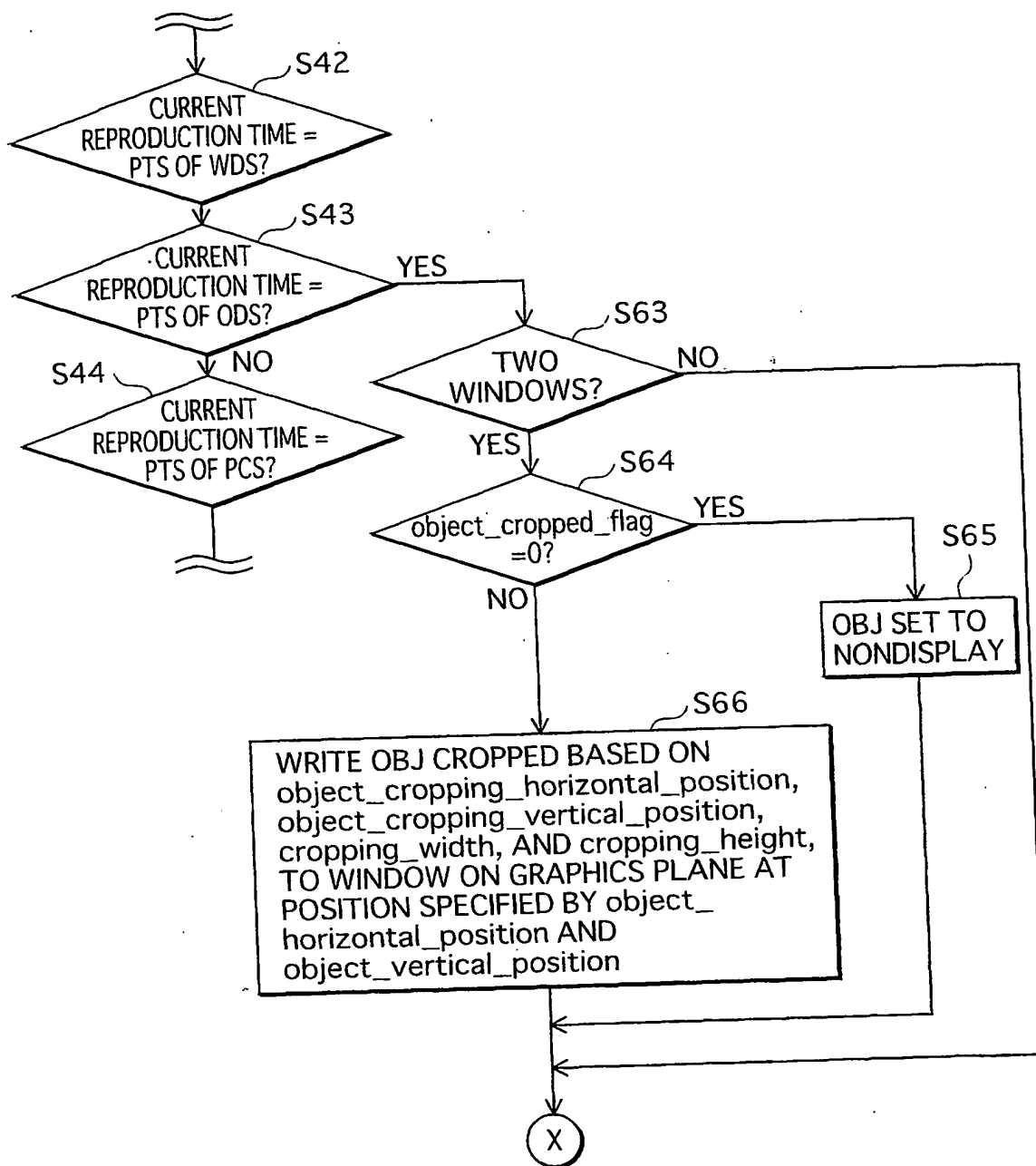


FIG. 41

